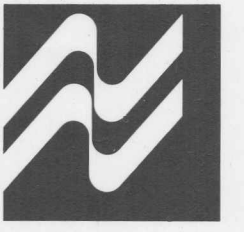
48-SERIES MICROPROCESSORS HANDBOOK

NATIONAL SEMICONDUCTOR

48-SERIES MICROPROCESSORS HANDBOOK

NATIONAL SEMICONDUCTOR



Preface

This handbook contains detailed design-related information pertaining to the National Semiconductor Corporation 48-Series single-chip microcomputers and microprocessors. The material presented is at a level of detail to aid in the design and development of systems using the 48-Series microcomputers. The information contained herein discusses the 48-Series architecture, expansion, and instruction set. Additionally, hardware examples integrated with the required software and data sheets of compatible devices are given.

The information contained in this handbook is accurate at the time of publication, but is subject to change without notice.

Additional copies of this or other National Semiconductor publications may be obtained from the National Semiconductor sales office or distributor serving your locality.

2900 Semiconductor Drive, Santa Clara, CA. 95051
(408) 737-5000/TWX (910) 339-9240

Table of Contents

Table of Contents (Cont'd.)

Appendix A

Appendix A (Cont'd.)

Appendix B

List of Illustrations

List of Illustrations (Cont'd.)

List of Tables

Chapter 1

The 48-Series Microcomputers

1.1 INTRODUCTION

The 48-Series microcomputers, designed using National's proprietary XMOS process, are true single-chip microcomputers. The devices making up this family of microcomputers contain all of the functions on one chip that are normally performed by multi-chip systems. The various chips that constitute the 48-Series are similar, only the amount of internal RAM and ROM varies. The features contained in the 48-Series are given below. Included within these features are several transparent improvements over other similar devices.

- 8-Bit CPU
- Built-in RAM and ROM (Externally Expandable)
- 1.36 μsec or 2.5 μsec Cycle Times
- Built-in Oscillator and Clock (Crystal-Controlled or External)
- 27 I/O Lines (Expandable)
- 8-Bit Timer/Counter
- Interrupt (Schmitt-Trigger with Hysteresis)
- 96 Instructions (70% Single-Byte)
- Single-Step
- Binary and BCD Arithmetic
- 8-Level Stack
- Programmable Standby RAM
- Low Voltage Standby (2.2V min.)
- On-Chip Standby Battery Charging

This expanded and improved 48-Series of microcomputers offers the user greater flexibility both during and after the development cycle. During development, three devices containing varying amounts of RAM, but no ROM, allow the user to develop an optimum system using as much external EPROM as is necessary.

Once the firmware is finalized, three mask-programmable devices, with varying amounts of RAM and ROM, can be substituted into the final system. The ability to substitute a single 48-Series device for multiple devices permits a low-cost, low-power alternative to potentially expensive applications. Additionally, future increased firmware needs are handled by simply upgrading to a device with a greater ROM capacity.

The varying amounts of internal RAM and ROM for the 48-Series microcomputers are given in *Table 1-1*.

Table 1-1. 48-Series Features

Device	ROM	RAM	48-Series Common Features
INS8035	None	64 bytes	8-bit CPU
INS8039	None	128 bytes	On-board Timing and Control
INS8040	None	256 bytes	27 I/O Lines
INS8048	1K bytes	64 bytes	Timer/Counter
INS8049	2K bytes	128 bytes	Interrupt
INS8050	4K bytes	256 bytes	Reset

Upgrading from an INS8049 to an INS8050 is a simple matter of removing one and replacing it with the other. All 48-Series devices are not only pin-compatible but are also software-compatible. With the many transparent improvements, and the option of operating a system from 1MHz to 11MHz, the 48-Series microcomputers are equal to or better than their equivalents.

In addition to the features found within the 48-Series microcomputers, these microcomputers are fully externally expandable. Not only can additional external RAM and ROM be added, but additional input/output lines can be added as well. Expansion in all cases is via standard memories or peripherals. With the variety obtainable within the 48-Series, the system designer can mix and match internal and external memory as well as peripherals to obtain an optimum system.

The 24 I/O ports on the 48-Series can be expanded up to 36 by adding an I/O expander, the INS8243, to one of the I/O ports. Multiple INS8243s can be bused together for systems with larger I/O requirements. An expanding array of memory and peripheral products permit creation of specific and specialized applications systems.

A list of 48-Series compatible components is given in *Table 1-2.*

Table 1-2. 48-Series Compatible Components

Device	Description
	Analog I/O Components
ADC080X	8-Bit Differential Input A/D Converters
ADC 3511/ ADC3711	3 1/2 - 3 3/4 Digit A/D Converters
Device	**Description**
	Communications Components
INS2651	Programmable Communications Interface
INS8250	Asynchronous Communications Element
DP7304B/ DP8304B	8-Bit Bidirectional Transceiver
Device	**Description**
	Digital I/O Components
INS8243	Input/Output Expander
DM7131/ DM8131	6-Bit Unified Bus Comparator
INS8202/03	Octal Buffers
INS8208	8-Bit Bidirectional Transceiver
INS8212	8-Bit Input/Output Port
INS8216/ INS8226	4-Bit Bidirectional Bus Transceivers
MM54C373/ MM74C373	Octal D-Type Latch
MM54C/374 MM74C374	Octal D-Type Flip-Flop
Device	**Description**
	Memory Components
MM2716	2K x 8 EPROM
MM2708	1K x 8 EPROM
MM54C920/ MM74C920	256 x 4 CMOS RAM
MM52116	2K x 8 ROM
MM52132	4K x 8 ROM
MM52164	8K x 8 ROM
Device	**Description**
	Peripheral Control Components
INS8253	Programmable Interval Timer
INS8350	Series Programmable CRT Controllers
INS8259	Programmable Interrupt Controller

The instruction set makes the 48-Series an efficient controller and arithmetic processor. With 96 instructions, 70% of which are one byte in length the remainder being two bytes, efficient use is made of the system memory. The I/O lines can be individually set, reset, or logically manipulated directly by the software. Additionally, a large set of branch and table look-up instructions provide for efficient logical operations. The internal timer/counter is also directly controlled through dedicated instructions.

1.2 PROGRAMMING REQUIREMENTS

Following final system design, it may be necessary for the initial production units to be shipped with EPROMs in the system. In order to reduce production costs to a minimum, the EPROMs should be replaced with mask-programmed 48-Series microcomputers. Conversion requires the user to send National Semiconductor a copy of the program, in paper tape or EPROM. National commits the program to mask, and the users 48-Series microcomputer is created. Additionally, the user-specified mask-programmable options may be designed into the system to create a unique, specialized controller.

Information on National's 48-Series programming requirements is contained within the data sheet in Appendix A.

Note: *There are some programming restrictions when using external ROM for program memory. The AND and OR to BUS instructions will not work with external ROM.*

1.3 DEVELOPMENT SUPPORT

48-Series product development is fully supported by National Semiconductor's STARPLEX™ Development System. STARPLEX is a general-purpose microcomputer/microprocessor development system with new levels of operating simplicity. As an interactive system, STARPLEX combines ease-of-use, through extensive use of prompts, with all the functions normally expected of a sophisticated development system. Included in the STARPLEX package are a full-function keyboard with a unique system resource keypad, a video monitor, 80 characters-per-line thermal printer, dual floppy disk subsystem, 64K bytes of RAM, and an 8080-based CPU board. The standard STARPLEX software package also includes a disk operating system, assembler, debugger, editor, linker, loader, FORTRAN, BASIC, and on-board ROM diagnostics and utilities. A cross-assembler for the 48-Series microcomputers is also available (as well as cross-assemblers for other microprocessors).

Two options available for STARPLEX are an in-system emulator (ISE) and a PROM programmer personality module. The ISE is a peripheral that permits STARPLEX itself to substitute for the product microprocessor. The PROM programmer module supports standard PRO-LOG-compatible personality modules. With these aids, the integration of system hardware and software is considerably eased.

Chapter 2

The 48-Series Single-Chip System

2.1 INTRODUCTION

As single-chip microcomputers, the 48-Series systems provide minimum chip implementations for controller applications. The different devices in the family are all functionally compatible. Each device contains a built-in system timing, control logic, 27 I/O lines, and differing amounts of RAM, and ROM. The RAM and ROM built into each device in the family is given in *Table 2-1*.

Table 2-1. RAM and ROM Implementation

DEVICE	RAM	ROM
INS8035	64 x 8	None
INS8039	128 x 8	None
INS8040	256 x 8	None
INS8048	64 x 8	1K x 8
INS8049	128 x 8	2K x 8
INS8050	256 x 8	4K x 8

The following sections contain descriptions of the various functional units of the 48-Series microcomputers.

2.2 48-SERIES ARCHITECTURE

There are nine major functional blocks comprising the 48-Series microcomputers. These blocks are in turn composed of various sub-blocks. The various blocks are as follows:

- CPU
  - — Instruction Register
  - — Arithmetic Logic Unit
  - — Accumulator
  - — Flag Register
- Branch Logic
- Resident ROM
  - — Program Counter
- Resident RAM
  - — RAM
  - — Registers
  - — Stacks
- Input/Output Ports and Bus
- Instruction Register
- Program Status Word
- Internal Timer/Counter
- On-Board Timing and Control Logic

A functional block diagram of the 48-Series microcomputers is shown in *Figure 2-1*.

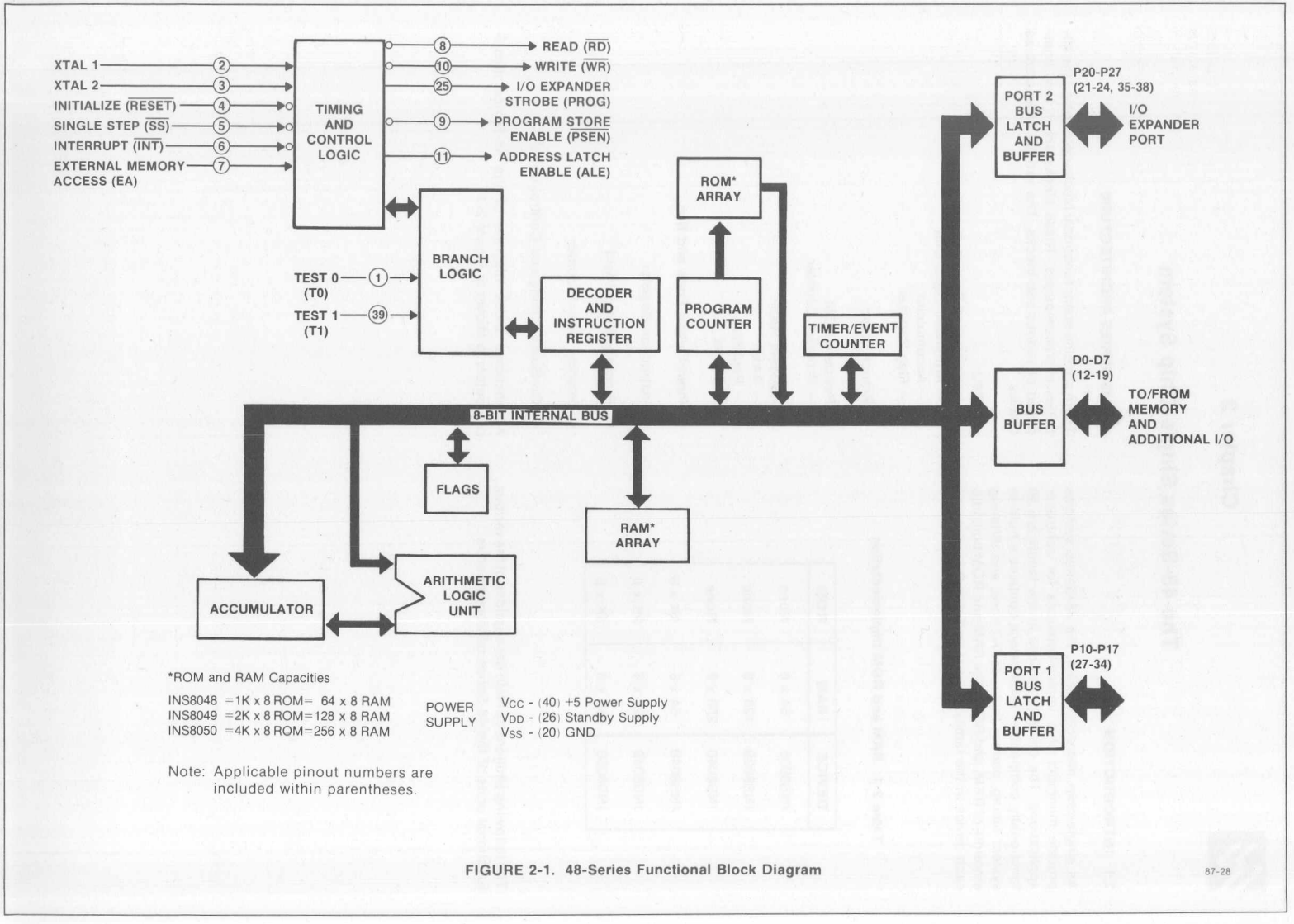


FIGURE 2-1. 48-Series Functional Block Diagram

87-28

2.2.1 CPU

The CPU is an 8-bit unit comprised of an instruction register, an arithmetic logic unit, an accumulator, and a flag register. In a typical operation, data placed in the accumulator combines with data from another source on the internal bus. The result of the combination is then stored in either the accumulator or a designated register.

Instruction Register: The instruction register receives the operation code (opcode) portion of each instruction taken from ROM. The opcode then generates specific outputs to control each block of the CPU. The outputs typically control the source and destination registers, as well as the function to be performed by the arithmetic logic unit.

Arithmetic Logic Unit: The arithmetic logic unit (ALU) operates on the 8-bit data taken from the accumulator and the bus. Operation is controlled by the instruction register and consists of the following operations:

- Add, With or Without Carry
- AND, OR, EXOR
- Decimal Adjust
- Increment/Decrement
- Complement, Clear
- Rotate Right and Left, With or Without Carry
- Swap Nibbles

Any operation performed in the ALU that results in values greater than eight bits, causes the carry flag to be set.

Accumulator: The accumulator is the main input port to the ALU. All operations are performed with reference to the accumulator. Most data transfers into the ALU from memory or the I/O ports pass through the accumulator.

The accumulator is a register normally used for 8-bit operations. As such, the accumulator maintains a carry bit immediately following bit 7. Additionally 4-bit binary-coded decimal (BCD) operations can also be performed by the accumulator. To support the 4-bit operations, an auxiliary carry bit is maintained immediately following bit 3. Auxiliary carry is affected by the same logical and arithmetic instructions that affect carry. Carry is the only testable bit. Auxiliary carry is used only when converting the accumulator contents from binary to BCD (using the DAA instruction). The auxiliary carry flag bit can be cleared by moving a zero into bit 6 of the program status word. An illustration of the accumulator is shown in *Figure 2-2.*

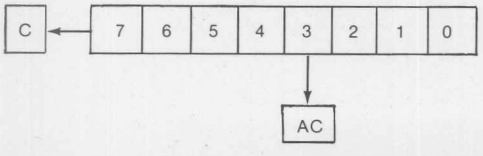


FIGURE 2-2. 48-Series Accumulator

2.2.2 Branch Logic

The branch logic within the 48-Series microcomputers permits the user to test various internal or external status conditions. If the selected condition is true, the branch logic forces a jump to the address specified by the program. The various branch condition tests are listed in *Table 2-2.*

Table 2-2. 48-Series Branch Conditions

Test	Logic Condition	Instruction
Interrupt*	0	JN1
Flag 0	1	JF0
Flag 1	1	JF1
Timer Flag	1	JTF
Carry	0 or 1	JC, JNC
Accumulator	0 or Non-0	JZ, JNZ
Accumulator Bit Test	1	JB0-JB7
Test 0*	0 or 1	JT0, JNT0
Test 1*	0 or 1	JT1, JNT1
Register**	Non-0	DJNZ

*External input
**Register can be tested for non-0 following a decrement

2.2.3 Program Status Word

The program status word is an 8-bit word stored in the program status register. The register contains both status information relating to machine operation, and the stack pointer. The contents of the register can be read from, or written to, the accumulator. All eight bits must be read or written at the same time.

During subroutine calls or interrupts, the upper four bits of the status register are saved on the stack. The contents may be restored to the register upon return, depending upon the return instruction used.

The stack pointer comprises the lower three bits of the program status word, and is an independent counter that points to designated spaces in the internal RAM.

The stack occupies RAM locations 8 through (X'17). When reset to zero, the stack pointer actually points to locations 8 and 9 in RAM. An interrupt or subroutine call causes the contents of the program counter and the upper 4 bits of the program status word to be stored in one of the eight stack registers. The stack pointer is then incremented to point to the next two locations.

Up to eight subroutines can be nested at any given time without the stack overflowing. Since the stack pointer is a simple up/down counter, an overflow will cause the deepest address to be lost (the counter overflows from 111 to 000). The pointer also underflows from 000 to 111.

Returns from subroutines decrement the stack pointer, with the contents of the register pair restored to the program counter and possibly the PSW. Depending upon the return instruction used, status may also be restored.

Note: *When the level of subroutine nesting is less than 8, the unused stack locations may be used as RAM.*

An illustration of internal stack composition is shown in *Figure 2-3*.

The program status word contents are given in *Table 2-3*.

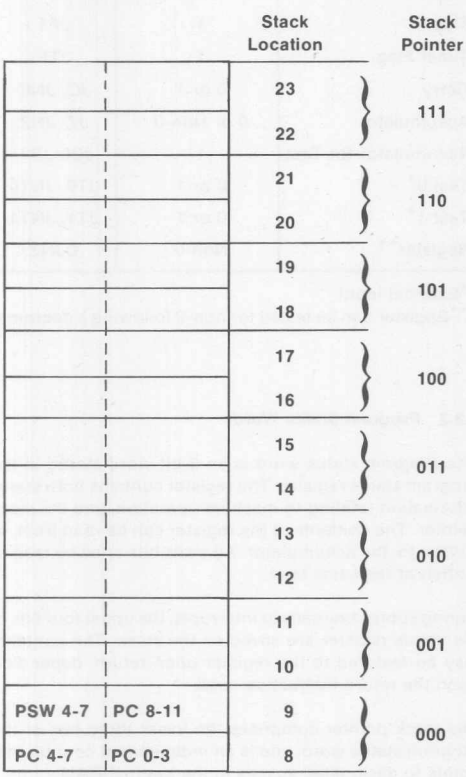


FIGURE 2-3. Internal Stack Compositon

Table 2-3. Program Status Word

Bit Position	Contents
0	Stack Pointer Bit, S0
1	Stack Pointer Bit, S1
2	Stack Pointer Bit, S2
3	Not Used. A logic 1 when read.
4*	Register Bank Select Bit 0 = Bank 0 1 = Bank 1
5*	Flag 0 (F0). A user controllable flag that can be complemented, cleared, or tested.
6*	Auxiliary Carry. A carry from bit 3 to bit 4 generated by an add operation. Used only by the decimal adjust instruction.
7*	Carry. A bit indicating the preceeding operation resulted in an overflow or an underflow.

Note: *Bits 4 through 7 saved on stack during subroutine calls or interrupts.*

2.2.4 Program Counter

The program counter is an independent 12-bit counter. For normal operation, the counter operates as a sequential up-counter, the output of which generates addresses for ROM. Bit 11 of the program counter is set independently of the normal count sequence by the memory bank-select instructions. In this manner, instruction fetches above or below the 2K memory boundary are effected.

During interrupts or subroutine calls, the contents of the program counter are stored in one of the eight selected stack locations (for additional information, see 2.2.3 and 2.2.6).

During external program fetches, the lower eight bits of the program counter are preset on the Bus port only during ALE, the upper four bits are held on Port 2. The thus addressed instruction is taken in on the Bus port when the program store enable ($\overline{PSEN}$) signal is active. The program counter is reset to zero (X'000) when the reset input ($\overline{RESET}$) goes active.

An illustration of the program counter is shown in *Figure 2-4.*

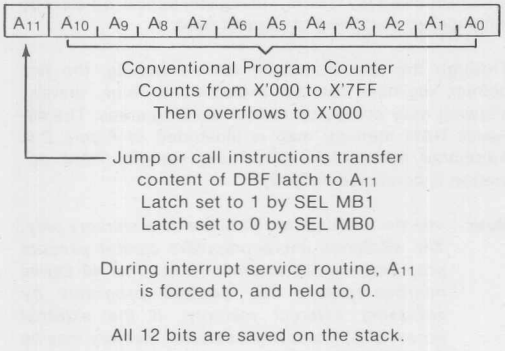


FIGURE 2-4. Program Counter

2.2.5 Resident ROM

The on-board ROM (for those devices containing ROM) is an 8-bit, mask-programmed ROM. Addressing the data or instructions within ROM is done by the program counter. Data, or instructions, output from ROM are placed onto the internal bus. ROM addressing, up to a maximum of 4K, is done through the 12-bit program counter. The INS8048 and INS8049 automatically address external memory when their internal memory boundaries of 1K and 2K, respectively, are crossed.

There are three dedicated addresses within ROM to provide for system initialization or branching. These three locations are described in *Table 2-4*. An illustration of the internal ROM organization is shown in *Figure 2-5*.

Table 2-4 Dedicated ROM Addresses

ADDRESS	FUNCTION
X'000	Reset. The reset input going low, forces the first instruction executed to be fetched from here.
X'003	Interrupt. The interrupt input going low (when interrupt is enabled), forces the first instruction of an interrupt service routine to be fetched from here.
X'007	Timer/Counter Interrupt. The timer/counter interrupt flag, when set (if timer/counter interrupt is enabled), forces the first instruction of a timer/counter service routine to be fetched from here.

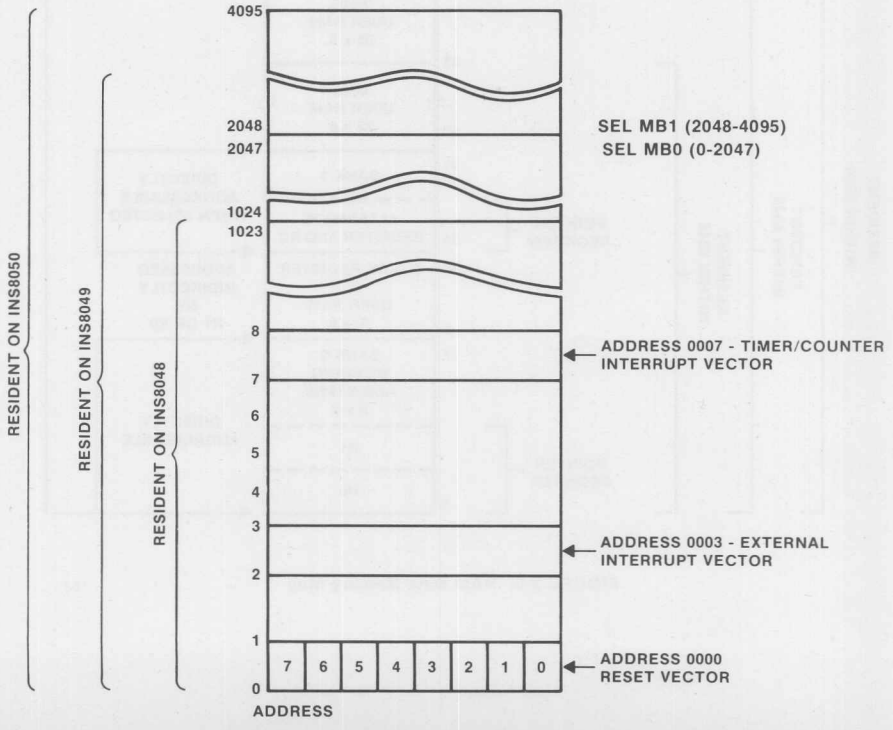


FIGURE 2-5. 48-Series ROM Memory Map 12-2

2.2.6 Resident RAM

The resident RAM data memory is arranged as 64, 128, or 256 bytes, depending upon the device. There are eight working registers in RAM for each register bank (RB0 and RB1) selected. Register bank 0 occupies 0 through 7, while register bank 1 occupies locations 24 through 31. Indirect addressing to all RAM locations is implemented through the two 8-bit pointer registers, R0 and R1. The pointer registers occupy the first two working register locations; 0 and 1 for register bank 0, or 24 and 25 for register bank 1.

The eight-level stack occupies the space between both working registers; locations 8 through 23. Each level of the stack actually occupies two memory locations. For additional information, see 2.2.3.

Register bank 1 and any unused stack locations may be used for RAM if the register bank-select feature and all of the stack are not used. Register bank 1 working registers may also be used as an extension of the register bank 0 registers, or reserved to service interrupts. The latter feature allows easy "saving" of register bank 0 registers by immediately switching to register bank 1.

Through the use of register bank-switching, the two pointer registers can be expanded to four, thereby allowing easy access up to four working areas. The 48-Series RAM memory map is illustrated in *Figure 2-6*. Additional information concerning standby RAM operation is contained in 2.4.2.

Note: *Internal and external RAM are data memory only. The 48-Series microcomputers cannot execute programs out of internal RAM. The 48-Series microcomputers can execute programs by accessing external memory. If the external access (EA) line is high, external memory may be ROM, PROM or RAM.*

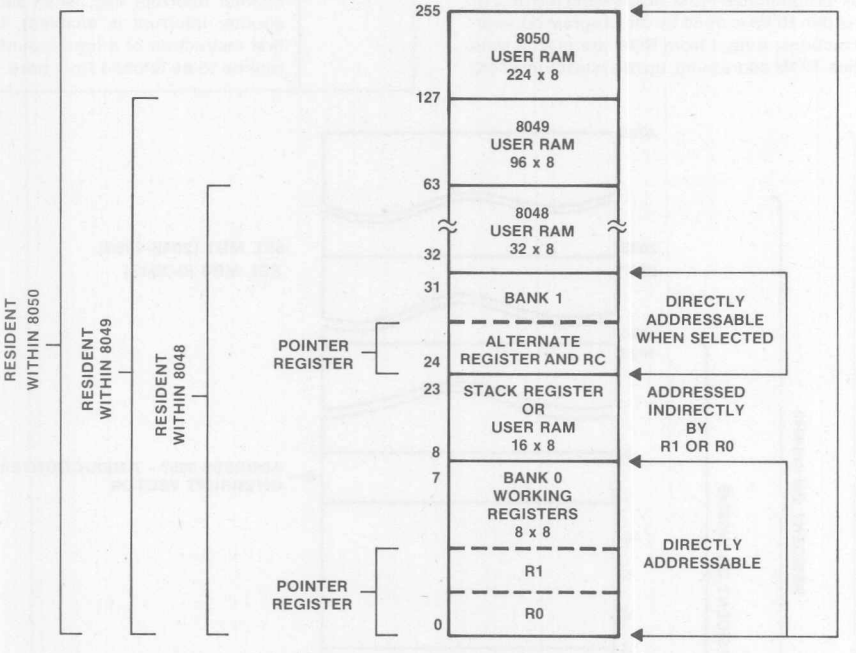


FIGURE 2-6. RAM Data Memory Map 12-3

2.2.7 Internal Timer/Counter

The 48-Series microcomputers contain an internal timer/counter that may be operated in two modes:

- As a timer
- As an event counter

These operating modes are independent of the CPU, thereby providing the facility of accurate time delays without tying up processor time.

The register for the timer/counter is an 8-bit, presettable up-counter. The register can be loaded or read using the MOV instructions to transfer data between the accumulator and the register.

Once started, the counter will count to its maximum value of X'FF, will overflow to X'00, and will continue counting until stopped by either a system reset or a stop instruction.

A system reset does not affect the contents of the register, although it will stop the counter. The only method of changing the register contents is to load the register using the MOV T instruction.

The counter may also be stopped using the STOP TCNT instruction. Once stopped, the counter remains stopped until either a start counter (STRT CNT) or start timer (STRT T) instruction is executed.

As the counter is incremented from X'FF to X'00, the timer overflow flag is set, and an unlatched interrupt request is generated. The flag may be tested by the JTF instruction. The request is honored only if EN TCNTI has been previously executed. The flag may be reset by either the JTF instruction or the Reset input going true.

Timer interrupt requests are stored in a latch, the output of which is ORed with the external interrupt. The timer interrupt can be enabled or disabled independently of the external interrupt. If the timer interrupt is enabled, an overflow will cause a subroutine call to location 7 in ROM, where a timer or counter service routine will start. If both an external interrupt and a timer interrupt should happen to occur simultaneously, the external interrupt is recognized first. Control will shift to location 3 to service the external interrupt. Since the timer interrupt is latched, once the external interrupt service routine is completed, control will return to the main program, at which time the timer interrupt will take effect. This time, control will shift to location 7 to service the timer interrupt. Once a subroutine call to location 7 occurs, the timer interrupt will be reset. The interrupt can also be reset by the DIS TCNT instruction. For additional interrupt information, see section 2.3.3.

Timer operation

For the timer/counter to operate as a timer, the STRT T instruction must be executed. Once executed, the instruction first clears and then causes the internal clock to pass through a divide-by-15 prescaler and a divide-by-32 prescaler. The second prescaler output increments the timer. By presetting the timer/counter prior to executing STRT T, accurate time-outs may be achieved.

As an example, assuming an 11MHz crystal is used, the input is divided by 15. The resulting 733KHz signal is in turn divided by 32. The final output of 22,917Hz increments the counter once every 44 microseconds.

By presetting the counter and detecting overflow, accurate timeouts between 44 microseconds and 11 milliseconds (256 counts) are possible. Timeouts longer than 11 milliseconds are possible by accumulating under software control, multiple overflows in one of the registers.

For times under 44 microseconds, the timer should be used as an event counter, with the external input taking the place of the internal clock source. Dividing the address latch enable signal (ALE) by three or more can serve as an external clock. Using the timer in this mode permits "fine-tuning" of timing delays through software looping.

Counter Operation

To operate the timer/counter as an event counter, the STRT CNT instruction must be used. High-to-low transistions on the T1 input will increment the counter. The counter can not be incremented any more than once per three instruction cycles. Dividing the address latch enable signal (ALE) by three or more provides a convenient source for this timing. The input at T1 must also remain high for 500 nanoseconds after each transition.

A functional block diagram of the timer/counter is illustrated in *Figure 2-7*.

2.2.8 Timing and Control Logic

The timing and control logic internal to the 48-Series microcomputers permits the following:

- External stimulus to control system operation
- System communication with external memory
- Generate clock signals for internal use

The various input and output signals that constitute the timing and control logic are listed in *Table 2-5*.

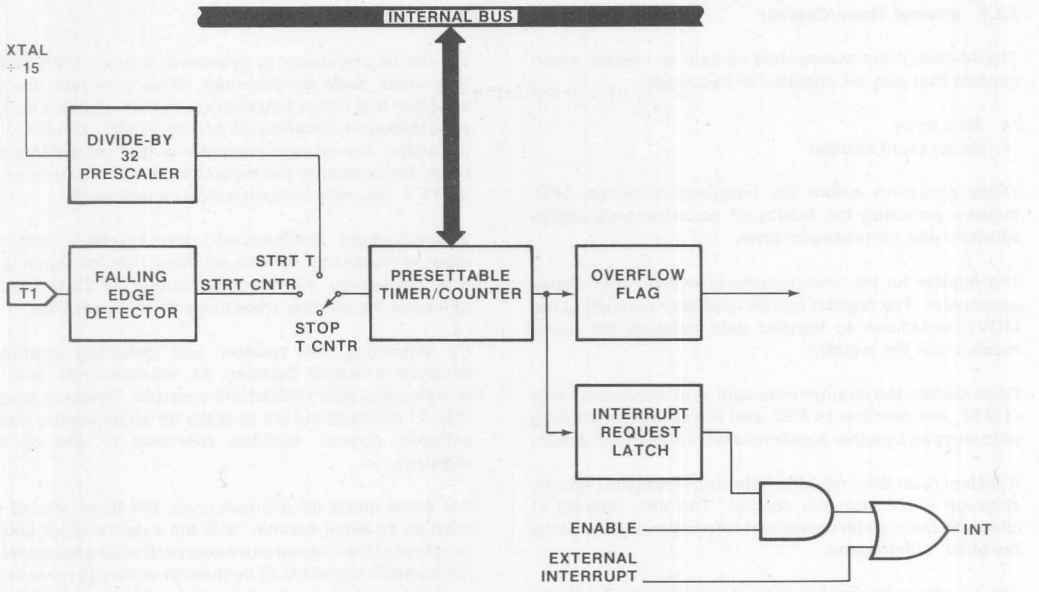


FIGURE 2-7. Timer/Counter Block Diagram 87-3

Table 2-5. Timing and Control Logic Signals

INPUTS	OUTPUTS
XTAL1, XTAL2 - Crystal, or Timing Source Inputs	RD - Read strobe
$\overline{\text{RESET}}$ - Initialize	$\overline{\text{WR}}$ - Write strobe
$\overline{\text{SS}}$ - Single Step	PROG - I/O expander strobe
$\overline{\text{INT}}$ - Interrupt	$\overline{\text{PSEN}}$ - Program store enable
EA - External access	ALE - Address latch enable

Internal Clock

The internal clock circuit of the 48-Series microprocessors accepts input from the two pins XTAL 1 and XTAL 2. A crystal or an externally generated clock source can be connected to these two inputs. The XTAL 1 pin (TTL compatible) is the input to a high-gain series-resonant circuit with a frequency range of 1 to 6MHz or 4 to 11MHz, depending upon the 48-Series part used. The XTAL 2 pin is the output of the circuit providing feedback to the crystal. If accurate frequency references and maximum speed are not required, an inductor can be used in place of the more accurate crystal.

The external clock frequency of the oscillator is divided by three to provide the basic clock cycle for the system. Each clock cycle comprises a single machine-state for the system. The basic clock cycle is available as an output at T0. Output is enabled by execution of the ENT0 instruction. Output is disabled whenever the system is reset.

There are five machine cycle times that comprise a single instruction cycle. Each of the five machine cycle times is, in turn, comprised of a single clock cycle. The address latch enable signal (ALE) is provided as a continual clock output to enable the 48-Series microcomputers to communicate with external memory. This signal is also derived from the five basic clock cycles that comprise a machine cycle.

A functional block diagram of the clock circuit is shown in *Figure 2-8*. Note that T0 is actually an input when the system is reset. Instruction cycle timing relationships are shown in *Figure 2-9*. An illustration of the basic timing relationships between the clock output from T0 and other system signals is shown in *Figure 2-10*. Instruction execution timing relationships are given in *Table 2-6*.

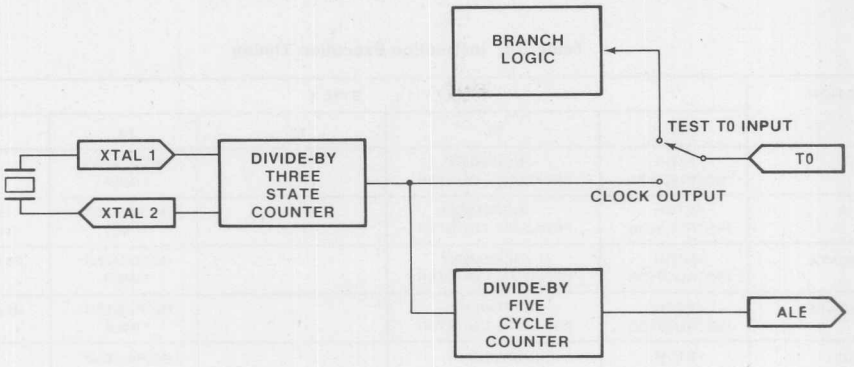


FIGURE 2-8. Internal Clock Block Diagram

87-4

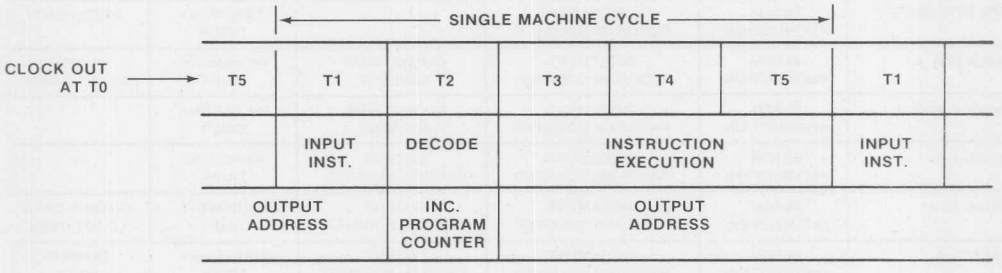


FIGURE 2-9. Instruction Cycle Relationships

87-5

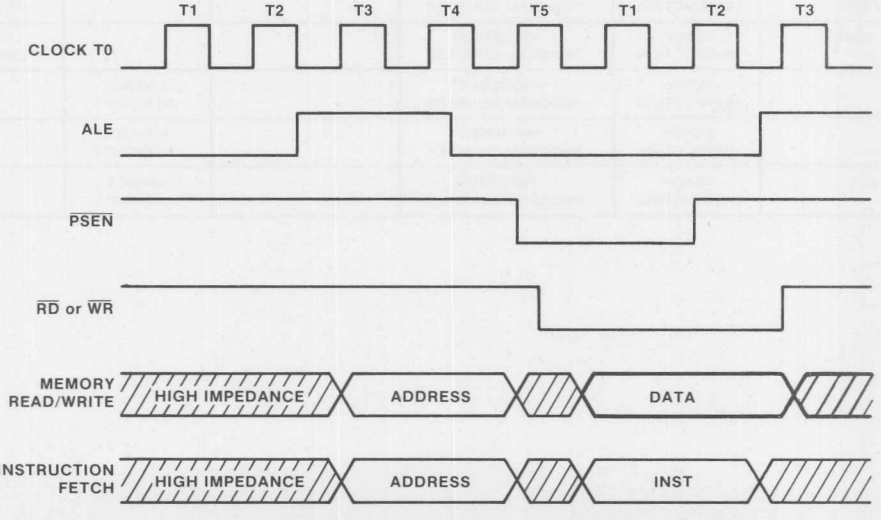


FIGURE 2-10. Timing Relationships

87-6

Table 2-6. Instruction Execution Timing

INSTRUCTION TYPE	BYTE 1				
	T1	T2	T3	T4	T5
IN A, P	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	—	INCREMENT TIMER	—
OUTL P, A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	—	INCREMENT TIMER	OUTPUT TO PORT
ANL P, #DATA	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	—	INCREMENT TIMER	READ PORT
ORL P, #DATA	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	—	INCREMENT TIMER	READ PORT
INS A, BUS	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	—	INCREMENT TIMER	—
OUTL BUS, A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	—	INCREMENT TIMER	OUTPUT TO PORT
ANL BUS, #DATA	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	—	INCREMENT TIMER	READ PORT
ORL BUS, #DATA	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	—	INCREMENT TIMER	READ PORT
MOVX @ R, A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT RAM ADDRESS	INCREMENT TIMER	OUTPUT DATA TO RAM
MOVX A, @ R	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT RAM ADDRESS	INCREMENT TIMER	—
MOVD A, P_i	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT OPCODE/ADDRESS	INCREMENT TIMER	—
MOVD P_i, A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT OPCODE/ADDRESS	INCREMENT TIMER	OUTPUT DATA TO P2 LOWER
ANLD P, A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT OPCODE/ADDRESS	INCREMENT TIMER	OUTPUT DATA
ORLD P, A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT OPCODE/ADDRESS	INCREMENT TIMER	OUTPUT DATA
J (CONDITIONAL)	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	SAMPLE CONDITION	INCREMENT TIMER	—
STRT T START CNT	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	—	—	START COUNTER
STOP TCNT	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	—	—	STOP COUNTER
ENI	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	—	ENABLE INTERRUPT	—
DIS I	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	—	DISABLE INTERRUPT	—
ENT0 CLK	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	—	ENABLE CLOCK	—

Table 2-6. Instruction Execution Timing (Cont'd.)

INSTRUCTION TYPE	BYTE 2				
	T1	T2	T3	T4	T5
IN A, P	—	READ PORT	* —	—	—
	—	—	* —	—	—
ANL P, #DATA	FETCH IMMEDIATE DATA	—	* INCREMENT PROGRAM COUNTER	OUTPUT TO PORT	—
ORL P, #DATA	FETCH IMMEDIATE DATA	—	* INCREMENT PROGRAM COUNTER	OUTPUT TO PORT	—
INS A, BUS	—	READ PORT	* —	—	—
	—	—	* —	—	—
ANL BUS #DATA	FETCH IMMEDIATE DATA	—	* INCREMENT PROGRAM COUNTER	OUTPUT TO PORT	—
ORL BUS, #DATA	FETCH IMMEDIATE DATA	—	* INCREMENT PROGRAM COUNTER	OUTPUT TO PORT	—
	—	—	* —	—	—
MOVX A, @ R	—	READ DATA	* —	—	—
MOVD A, P_i	—	READ P2 LOWER	* —	—	—
	—	—	* —	—	—
	—	—	* —	—	—
	—	—	* —	—	—
J (CONDITIONAL)	FETCH IMMEDIATE DATA	—	* UPDATE PROGRAM COUNTER	—	—

*If external ROM is being accessed, valid instruction addresses are output at this time.

2.3 INPUT/OUTPUT SIGNAL DESCRIPTIONS

All pins on the 48-Series microprocessors, with the exception of the power and clock inputs, are input or output lines. The following sections describe in some detail the operation of these lines.

An illustration of the pin configuration for the 48-Series microcomputers is shown in *Figure 2-11*. A summary of the pin functions is given in *Table 2-7*.

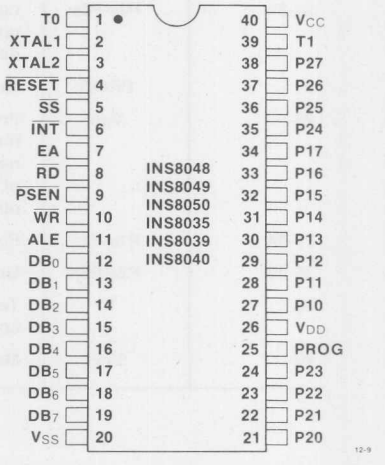


FIGURE 2-11. 48-Series Pin Configuration

Table 2-7. 48-Series Pin Summary

Pin	Designation	Function
1	T0	Testable input using JT0 and JNT0 instructions. Clock output once ENT0 CLK instruction is executed. Restored to an input line by reset.
2	XTAL 1	Crystal input for internal oscillator. Also input for external clock source.
3	XTAL 2	Crystal input for internal oscillator.
4	$\overline{\text{RESET}}$	Reset input for CPU. Active low.
5	$\overline{\text{SS}}$	Single step input. Used in conjunction with ALE to single step through program execution. Active low.
6	$\overline{\text{INT}}$	Interrupt input. Generates an interrupt if interrupt is enabled. Disabled after a reset. Active low.
7	EA	External Access, when taken high, forces all instruction fetches to come from external ROM.
8	$\overline{\text{RD}}$	Read strobe. Active during bus reads. Can enable data onto bus from external devices. Active low.
9	$\overline{\text{PSEN}}$	Program store enable. Active during instruction fetches from external ROM. Active low.
10	$\overline{\text{WR}}$	Write strobe. Active during bus writes. Can strobe data into external devices from bus. Active low.
11	ALE	Address latch enable occurs once during each machine cycle. Also useful as a clock output. The falling edge of ALE strobes address into external RAM and ROM.
12-19	DB_0-DB_7	Bus port. Bidirectional port which can be written or read synchronously, using $\overline{\text{WR}}$ or $\overline{\text{RD}}$. Output can also be statically latched.
		Outputs eight low-order address bits during external instruction fetches. Receives addressed instruction during $\overline{\text{PSEN}}$. Also passes eight address and data bits from external RAM (under control ALE, $\overline{\text{RD}}$, and $\overline{\text{WR}}$).
20	V_{SS}	Circuit ground.
21-24	$P2_0$-$P2_3$	Lower four bits of quasi-bidirectional Port 2. Outputs upper four address bits during external ROM access and data for INS8243 port expander.
25	PROG	Output strobe for INS8243 I/O expander.
26	V_{DD}	Provides an input for a standby power source for the internal RAM. When a standby power source is connected, V_{CC} provides a battery charging path. The amount of internal RAM connected to V_{DD} is a programmable option.
27-34	$P1_0$-$P1_7$	Port 1, quasi-bidirectional.
35-38	$P2_4$-$P2_7$	Upper four bits of quasi-bidirectional Port 2.
39	T1	Testable input using JT1 and JNT1 instructions. Event counter input once STRT CNT instruction is executed.
40	V_{CC}	Main 48-Series power source (+5V).

2.3.1 Reset ($\overline{\text{RESET}}$)

The reset input initializes the processor. The input is a Schmitt trigger that has an optional internal pullup resistor. An external 1-microfarad capacitor tied to this pin assures the reset input will be low as V_{CC} is brought high. If an external reset is used, the input must be held at ground for at least 50 milliseconds after the power supplies have stabilized. If the power and oscillator have already stabilized, the external reset need only be negative true for five machine cycle times. A list of internal functions reset by this signal going low is given below:

- Program counter set to zero.
- Stack pointer set to zero.
- Memory bank 0 and register bank 0 selected.
- Bus set to TRI-STATE mode, (except when EA is true)
- Ports 1 and 2 to input mode.
- Interrupts disabled (Timer/Counter and External).
- Timer stopped.
- Timer flag cleared.
- F0 and F1 cleared.
- Clock output at T0 disabled; T0 becomes testable input.

Two typical reset circuits are illustrated in *Figure 2-12*. For additional information, see 2.4.3.

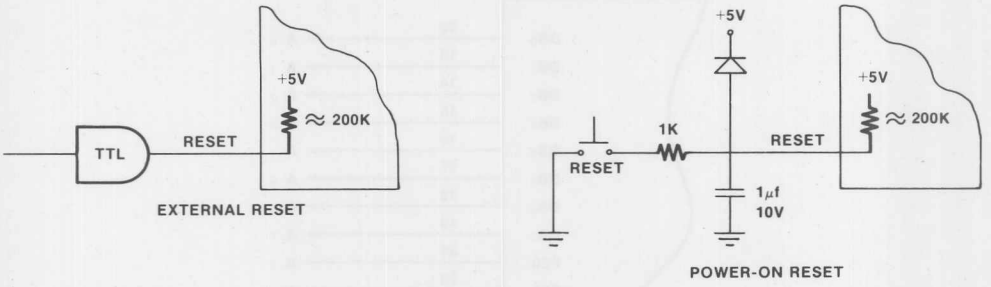


FIGURE 2-12. 48-Series Reset Circuits

87-7

2.3.2 Single Step ($\overline{\text{SS}}$)

The single step input, in conjunction with the ALE strobe, allows the processor to step through ROM, logically performing one instruction at a time. If the instruction is two cycles in length, both cycles are executed and the processor then stops. Instruction execution can be easily followed as the address of the next instruction to be executed is output on both Bus and the lower 4 bits of Port 2. The Bus buffer contents are lost while the processor is stopped. If necessary, the Bus data can be latched externally on the leading edge of ALE and thus saved.

The single step operation is given in the following steps. Reference should also be made to the timing chart of *Figure 2-13*.

1. Single step goes low, requesting the processor to stop.
2. The processor stops during the instruction fetch of the next command. If a double-byte instruction, both bytes are taken, following which the processor stops. ALE goes high as an acknowledgement.
3. At this point, the address taken from the program counter is output on both the Bus and Port 2. This state can last indefinitely. Bus and port bit composition is illustrated in *Figure 2-14*.
4. When single step goes high, the processor is allowed to continue.
5. The processor acknowledges by driving ALE low.
6. In order to stop the processor at the next instruction, $\overline{\text{SS}}$ must be driven low immediately after ALE goes low. If $\overline{\text{SS}}$ remains high, the processor will continue to run.

Implementation of a single-step circuit for the 48-Series requires a minimum of components. An illustration of such a circuit is shown in *Figure 2-15*. Circuit operation is as follows:

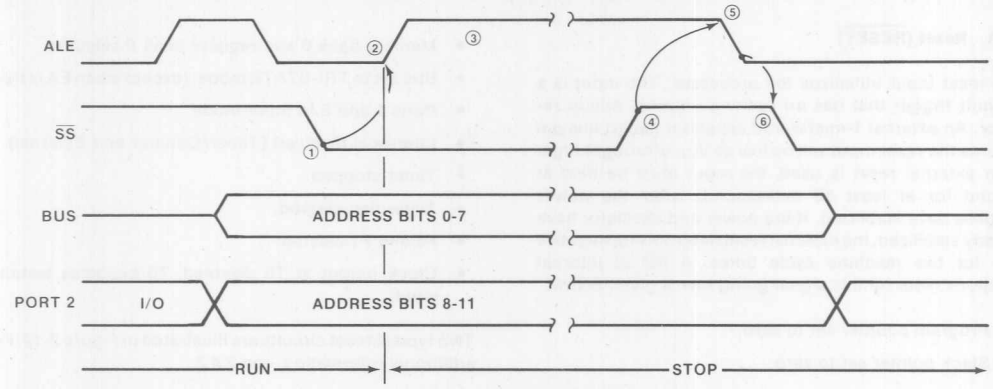


FIGURE 2-13. Single Step Timing 87-8

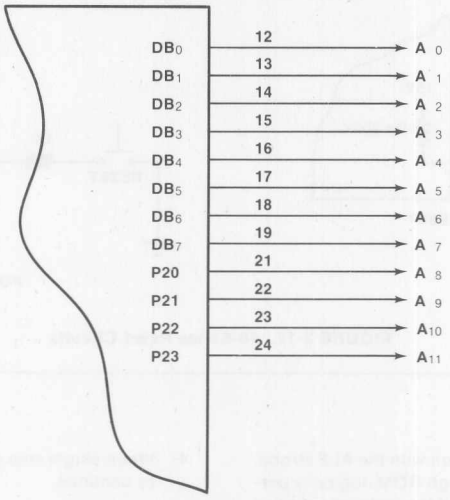


FIGURE 2-14. Address Output Lines 87-9

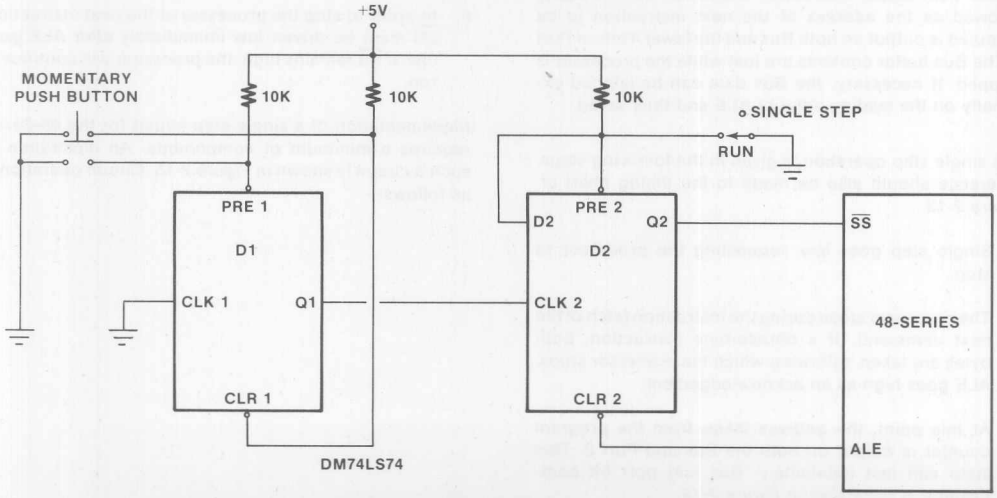


FIGURE 2-15. Single Step Circuit 87-10

During normal operation the preset input (PRE2) of the DM74LS74 is held low, forcing the Q2 output high, thereby holding $\overline{SS}$ high. As long as $\overline{SS}$ remains high, the computer will continue to run. Switching the run/single-step switch to the single-step position puts a logic 1 on PRE2. As soon as ALE goes low, $\overline{SS}$ will be forced low, causing the computer to stop. The next instruction is executed by pressing the momentary switch. If ALE is high, the debounce flip-flop (D1) clocks a 1 into D2. As soon as $\overline{SS}$ goes high, the computer fetches another instruction that brings ALE low. When ALE goes low, D2 is cleared, $\overline{SS}$ goes low, and the computer stops. Placing the run/single-step switch back in the run position forces a 1 to SS, allowing the computer to resume execution.

2.3.3 Interrupt ($\overline{INT}$)

The interrupt input, when enabled, will initiate an interrupt. This input has a Schmitt trigger input, with hysterisis, that is active low. If the interrupts are disabled, the $\overline{INT}$ line can still be sampled by a conditional jump instruction. When an interrupt is detected, a jump-to-subroutine at location 3 (in internal ROM) occurs as soon as the current instruction completes execution.

During interrupts, the program counter and upper four bits of the program status word are stored on the stack. For additional information see section 2.2.3. Control is transferred to location 3 in ROM, which should normally contain a jump to interrupt service routine. Once completed, the interrupt service routine should end with a return-and-restore-status instruction (RETR).

Being a single-level interrupt, $\overline{INT}$ is disabled while servicing an interrupt, and is re-enabled by the RETR instruction. Internal timer/counter interrupts are treated in like manner. If both an external and a timer/counter interrupt occur at the same time, the external interrupt has priority. Additional information on timer interrupts is given in section 2.2.8.

A simple programming trick allows the timer/counter to function as a second external interrupt, if desired. By loading X'FF in the timer and putting the timer/counter in the event counter mode, a logic 1-to-logic 0 transition at input T1 will cause the counter to increment and overflow creating an interrupt vector to location 7 in ROM.

Interrupts are disabled by either system reset or the disable interrupt instructions: DIS I for an external interrupt, DIS TCNTI for a timer/counter interrupt. The interrupts must be enabled by the program for them to function; ENI for an external interrupt, EN TCNTI for a timer/counter interrupt. Interrupt sampling occurs each machine cycle during ALE. An interrupt request must be removed before ending the interrupt service routine. If not removed, the processor will immediately re-enter the service routine. A selected output line from the 48-Series microcomputers could be designated an interrupt-acknowledge line. This line can then be activated during the interrupt service routine to reset the requesting interrupt.

The $\overline{INT}$ input may also be tested by the jump-if-interrupt-is-low instruction (JNI). If $\overline{INT}$ is left disabled, this input can be tested in the same manner as inputs T0 and T1. An illustration of the internal interrupt structure is shown in *Figure 2-16.* Additional information is given in 2.2.8 and 3.2.2.

2.3.4 External Access (EA)

When the external access input is driven high, the microcomputer performs all memory fetches from external ROM, internal ROM is disabled. Normal 48-Series usage would have the user program stored in internal ROM. As an example, diagnostic routines to test the internal logic of the CPU could be contained in the external ROM. EA should be driven high only while $\overline{RESET}$ is low. For the INS8035, INS8039 and INS8040, EA should be tied to V_{CC}. For additional information, see 3.2.

The internal ROM may also be read, independent of the CPU, using the EA input. The sequence for reading internal memory is as follows:

- $\overline{RESET}$ is driven to 0 volts.
- CPU is placed in the read mode by driving EA to +12 volts.
- Address to be read is placed at the BUS and Port 2 lines (P20 through P22).
- $\overline{RESET}$ goes high, at which time the addresses are latched.
- After the addresses are latched, $\overline{RESET}$ remaining high will cause the contents of the addressed location to be output on the BUS lines.

Timing for reading internal ROM is shown in *Figure 2-17.*

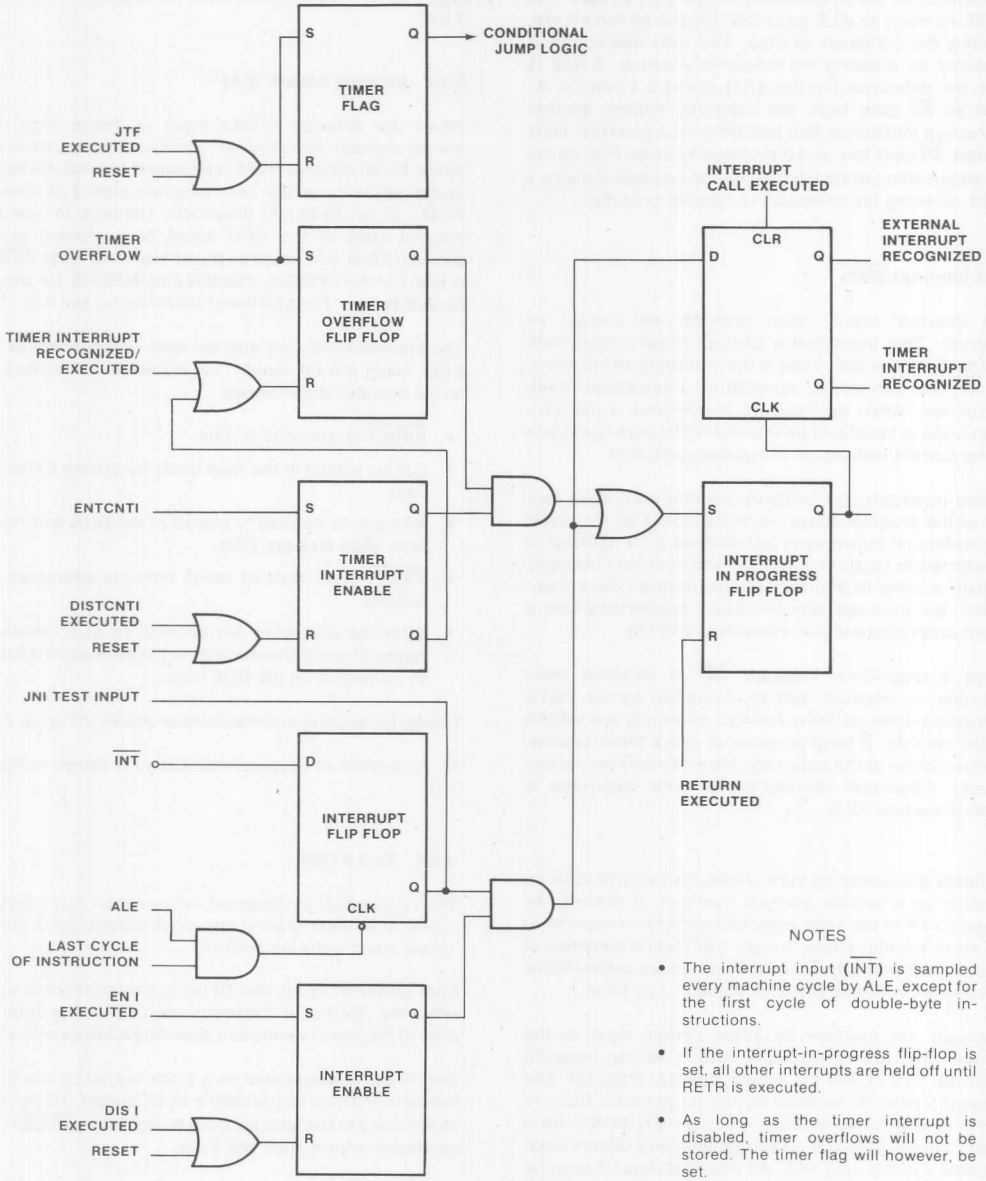
An illustration of a typical read circuit is shown in *Figure 2-18.*

2.3.5 Test 0 (T0)

Pin T0 is a dual purpose pin - in one state it is a testable input, in another state it is a clock output. Each state is under direct software control.

After power-on reset, the T0 pin is a testable input using either the JNT0 or JT0 instructions. Depending upon the state of the input, a jump to a specified address will occur.

The T0 pin is designated as a clock output by the ENT0 instruction. Once designated a clock output, T0 can only be returned to the testable state by activating $\overline{RESET}$. For additional information, see 2.2.8.



NOTES

- The interrupt input ($\overline{INT}$) is sampled every machine cycle by ALE, except for the first cycle of double-byte instructions.
- If the interrupt-in-progress flip-flop is set, all other interrupts are held off until RETR is executed.
- As long as the timer interrupt is disabled, timer overflows will not be stored. The timer flag will however, be set.

FIGURE 2-16. Internal Interrupt Structure

87-11

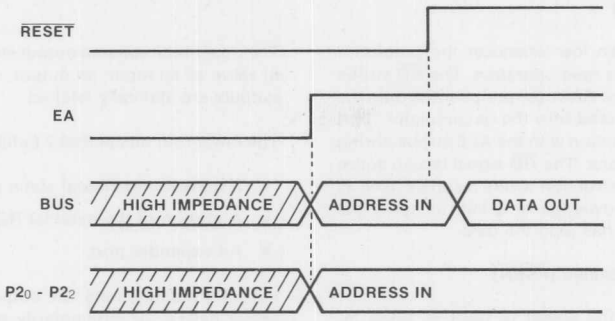


FIGURE 2-17. Timing for Reading Internal ROM 87-12

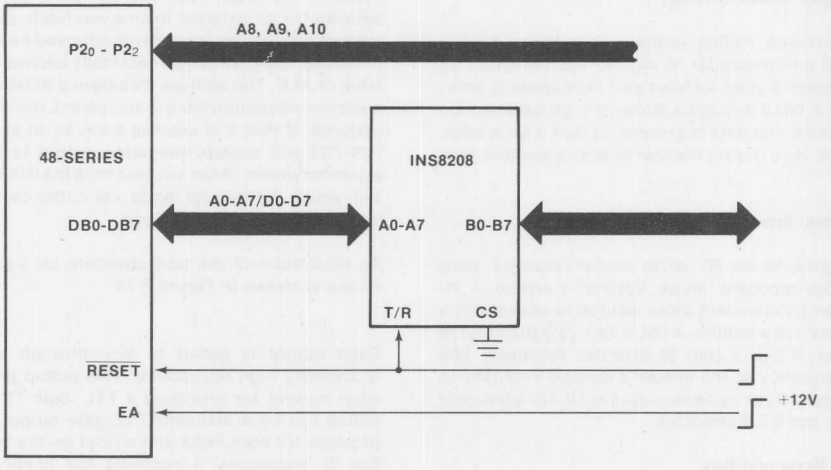


FIGURE 2-18. Reading Internal ROM 87-13

2.3.6 Test 1 (T1)

Pin T1 is also a dual purpose pin — in one state it is a testable input, in another state it is the input to the event counter. For each case, the input function is software controllable. As a testable input, T1 is tested by either the JNT1 or JT1 instructions, with a jump to a specified address occurring if the tested state is true.

The T1 pin is designated as the event counter input by the STRT CNT instruction. Thereafter, high-to-low transitions at T1 will increment the counter. For additional information, see 2.2.7.

2.3.7 Address Latch Enable (ALE)

The address latch enable strobes permit external latching of the address bits present on the Bus outputs and Port 2, bits 0 through 3. The ALE signal occurs once each machine cycle, with the address valid on the falling edge of ALE. The signal ALE may also be used as a clock signal to enable or disable the single step input. Address latch enable is used in conjunction with the program store enable, for external fetches, and the read and write strobes for data accesses from RAM or peripherals. For additional information, see 2.2.8 and 2.3.2.

2.3.8 Write Strobe ($\overline{WR}$)

The write strobe is driven low whenever the processor performs an external Bus write operation. Data output occurs on the Bus port, with the $\overline{WR}$ strobe writing the data into external RAM. This signal operates in conjunction with the ALE strobe during MOVX @R,A instructions. The $\overline{WR}$ signal is also active during an OUTL BUS A instruction where it can be used to notify a peripheral that new Bus port data is available.

2.3.9 Read Strobe ($\overline{RD}$)

The read strobe is driven low whenever the processor performs an external Bus read operation. The $\overline{RD}$ strobe enables data from external RAM, (or peripherals) onto the Bus, at which time it is read into the accumulator. This signal operates in conjunction with the ALE strobe during MOVX A, @ Rr instructions. The $\overline{RD}$ signal is also active during an INS A, BUS instruction where it can be used as either an interrupt acknowledge or a flag to notify the peripheral that the CPU has read the data.

2.3.10 Program Store Enable ($\overline{PSEN}$)

The program store enable signal is used to fetch instructions for the CPU. The $\overline{PSEN}$ strobe is active only when fetching instructions from external ROM. This signal operates in conjunction with the ALE strobe.

2.3.11 Output Strobe (PROG)

The output strobe, PROG, is the output strobe for the INS8243 I/O port expander. A high-to-low transition on PROG indicates that an address and instruction is available on Port 2, bits 0 through 3. A low-to-high transition on PROG indicates that data is present on Port 2 for a write. Data must be valid during the low-to-high transition for a read.

2.3.12 Crystal Input (XTAL 1, XTAL 2)

These two pins on the 48-Series microcomputers allow either a series-resonant device, such as a crystal or inductor, or an independent clock source to connect to a high-gain internal amplifier. XTAL 1 (pin 2) is the input to the amplifier; XTAL 2 (pin 3) provides feedback. The resonant frequency of the circuit is divided internally to create the basic clock cycle as output at T0. For additional information, see 2.2.8 and 2.3.7.

2.3.13 I/O Ports and Bus

There are 24 input/output lines and three test inputs in the 48-Series microcomputers. The I/O lines are organized as three 8-bit ports. The ports can be either input, output, or bidirectional. Ports 1 and 2 are especially versatile in that different types of outputs may be intermixed.

Ports 1 and 2 differ from Port 0 (the Bus), in that they are quasi-bidirectional, while the Bus is a true bidirectional port. This means that they can be used as inputs or outputs while being statically latched. To further explain this, if a 1 is written into any bit of Ports 1 or 2, that bit can function as an input or a high-level output. If a 0 is written into any of these bits, that bit can only function as a low-level output. This type of I/O pin is better understood as an open-drain output with a large value internal pullup resistor connected to an input latch. Data is latched in these ports from the CPU and will remain latched until changed. As inputs, these ports are non-latching, and must be read by an input instruction prior to removing the input.

The quasi-bidirectional output structure permits each line to serve as an input, an output, or both, even though the outputs are statically latched.

The lower four bits of Port 2 fulfill three distinct functions:

- A quasi-bidirectional static port.
- A portion of the external ROM address.
- An expander port.

For all three functions, the outputs are driven low by an active device, or momentarily pulled high by an active device; then held high by a passive device.

This port may contain latched output data yet still be used in another mode without affecting operation of either mode. If the lower four bits of Port 2 outputs are the address for an external instruction fetch, the previously latched data will be temporarily removed from the output. If needed, the data can be externally latched on the rising edge of ALE. The address for external ROM is output and once the instruction fetch is completed, the latched data is restored. If Port 2 is used as a bus to an expander port, P20-P23 will contain the value output to the INS8243 expander device. After an input from the INS8243, the port will remain in the input mode. For either case, previously latched data will be destroyed.

An illustration of the port structure for a standard TTL output is shown in *Figure 2-19*.

Each output is pulled to V_{CC} through a resistor of moderately high impedance. This pullup provides sufficient current for providing a TTL logic "1", yet can be pulled low by a standard TTL gate output. This ability provides for both input and output on the same pin. For fast '0' transitions, a relatively low impedance device (≈5K) is switched on momentarily when a 1 is written to the line. When a 0 is written to the line, a low impedance device (≈300Ω) overcomes the 5K pullup and provides TTL current sink capability.

Note: *Since the pulldown device is such a low impedance, a "1" must be written to the line prior to any input. A system reset ($\overline{RESET}$ going true) drives all lines to a high impedance "1".*

Port 1 and Port 2, when used with the ANL and ORL instructions, provide an efficient means for handling single-line inputs and outputs.

The Bus port is a true bidirectional port that can be statically latched or used synchronously. When used as a bus, valid input or output must occur during the read or write strobes. During external instruction fetches, the eight low order address bits from the program counter are preset at this port. The fetched instruction must be present at the input when the program strobe ($\overline{PSEN}$) goes high. During external RAM operations, addresses and data are output from this port under control of address latch enable (ALE) and the read ($\overline{RD}$) or write ($\overline{WR}$) strobes. When not being written to or read from, the Bus lines are in their high impedance mode (unless previously latched).

As a static output port, data is written and latched using the OUTL instruction. Data is input using the INS instruction. Both instructions generate pulses on the corresponding $\overline{WR}$ and $\overline{RD}$ output strobe lines (OUTL generates a $\overline{WR}$ pulse; INS generates a $\overline{RD}$ pulse).

The Bus port is not bit-programmable as are the other two ports (P1 and P2). Once written to, using an OUTL instruction, the bus will remain an output port until either the device is reset or the bus is used for an external memory access.

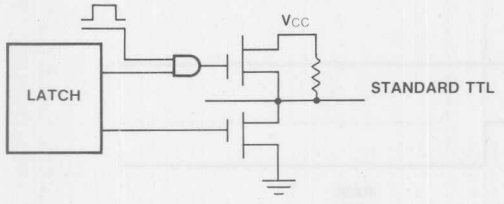


FIGURE 2-19. 48-Series I/O Port Options

87-14

2.4 SPECIAL FEATURES

All 48-Series microcomputers contain additional features not included in other similar devices. These features are totally transparent to the user if 48-Series microcomputers are substituted for the other devices - there is no adverse effect upon system operation.

These added features are:

- Schmitt trigger (with hysteresis) input for interrupt.
- Built-in battery charging circuit.
- Varying amounts of standby RAM (mask-programmable option) can alter standby current requirements.
- Mask-programmable pullup resistors on certain inputs.

2.5 BATTERY CHARGING CIRCUIT

The internal battery charging circuit is simply a solid-state switch between V_{CC}, an internal resistor, and V_{DD}. During normal operation, $\overline{RESET}$ is high, holding the switch in the closed state, thereby providing power for the internal RAM. The V_{CC} supply also provides the charging current for the external battery.

In the event of a power failure, $\overline{RESET}$ must go low before V_{CC} drops below 4.5V. By going low, $\overline{RESET}$ inhibits any RAM access and opens the internal switch. As soon as the switch is opened, current flow reverses and what was originally a charging output for the batteries, becomes a source of standby power for the internal RAM. Being an XMOS device, the internal RAM only requires a V_{DD} of 2.2 volts (2 Ni-Cad cells) to sustain data. The built-in charging circuit, and low standby voltage requirement, eliminates the need for both an external charging circuit and five Ni-Cad cells. An illustration of the internal charging circuit is shown in *Figure 2-20*.

A typical sequence of events leading up to a standby mode of operation would be as follows:

- An imminent power supply failure is detected.
- An interrupt is generated to vector operation to a standby service routine.
- All important data and status is stored in RAM.
- $\overline{RESET}$ goes low, inhibiting any RAM access and places RAM on standby power.

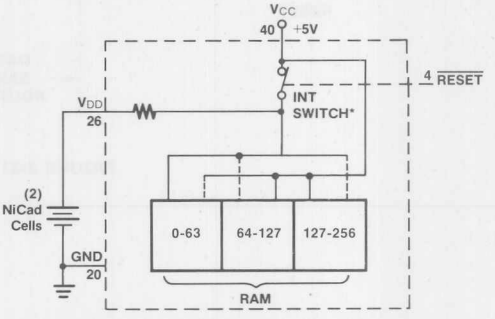


FIGURE 2-20. Internal Charging/Standby Circuit 12-5

Power supply failure-detection is the most critical portion of the standby operation. It is important that an imminent failure be detected in time to save all critical data or status.

The simplest and most effective detection circuit is a voltage comparator that monitors the DC supply. The battery back-up can supply the reference voltage while the scaled-down, unregulated portion of the supply provides the test voltage. Since DC regulators typically fall out of regulation when their input approaches within 2 volts of the output, the comparator can sense this voltage change long before the regulated output drops.

An imminent power failure generates an interrupt causing program operation to branch to a status-save routine. The status-save routine places all data critical to system operation in the standby RAM locations, thereby assuring continuation of the main program when power is restored.

System operation is such that once an interrupt is generated, the interrupt input is disabled until re-enabled by the enable external interrupt **(ENI)** instruction. Saving of status is thus assured while preventing multiple interrupts from a fluctuating power supply.

The reset input should be driven low by the firmware to prevent undesirable system operation during power down. It is important that the state of the power supply be checked again before issuing a reset to the microprocessor. The jump-if-negative interrupt **(JNI)** instruction can recheck the interrupt input without creating an interrupt. If the power supply only fluctuated slightly, this test permits the program to jump around a firmware reset and restore normal program operation. If the power fail signal is still active at the interrupt input, the reset input should be driven low. An effective method is to drive $\overline{\text{RESET}}$ low under firmware control as the final task of the interrupt service routine. If one of the output port bits is used to gate the interrupt into the reset input, the microprocessor itself can drive $\overline{\text{RESET}}$ low.

Note: *Since $\overline{\text{RESET}}$ sets all ports high, an active high input should be used to gate INT into $\overline{\text{RESET}}$, otherwise, $\overline{\text{RESET}}$ disappears immediately following execution of the instruction that caused it. This also means the bit must be initialized to a zero as part of the power-on sequence.*

If power is in fact removed, the next power-on reset forces the first instruction to be fetched from location 0. Therefore, it is the responsibility of the initialization sequence to determine whether or not power is being applied for the first time or being reapplied following a power loss. If it is a reapplication of power, program status must be restored and control returned to the interrupted program.

Additional information on the $\overline{\text{RESET}}$ function is given in section 2.3.1. An illustration of typical standby sequence timing is shown in *Figure 2-21*.

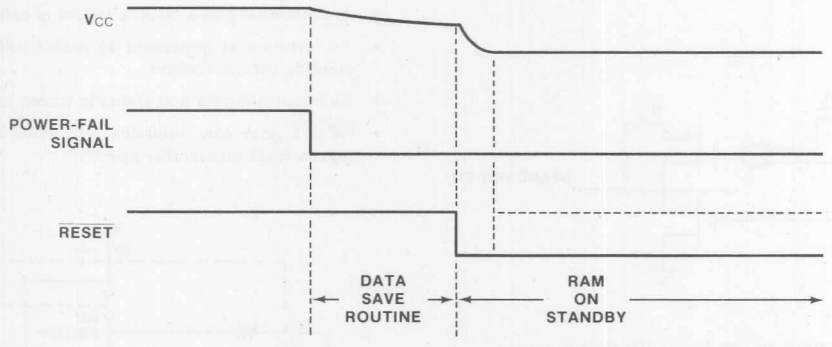


FIGURE 2-21. Standby Sequence Timing 87-15

Chapter 3

Expanding The 48-Series Microcomputers

3.1 INTRODUCTION

The expansion capabilities of the 48-Series devices offer the user great flexibility in both development and end applications.

The 48-Series devices offer varying amounts of internal RAM and ROM to allow simple expansion. Internal ROM varies from none up to 4K, while internal RAM varies from none up to 256 bytes. Additionally, three I/O ports are standard, with one port expandable allowing much larger systems if more than one chip is used.

Given the wide variety of 48-Series devices, development programs can be tested, using up to 4K of external EPROM. Once completed, the application can be mask-programmed into a single device for use.

NOTE: *AND and OR to BUS instructions will not work with external ROM.*

Depending upon how an application may grow in terms of memory requirements, different 48-Series devices can meet these expansion needs. If, for example, an application program should grow beyond the standard internal 2K ROM memory limit, the INS8050 with 4K of ROM, may be substituted. Not only are no additional external memory components required, but potentially higher power needs are eliminated.

The following sections describe in some detail the manner in which the 48-Series microcomputers can be expanded. Examples are given for added ROM, RAM, and I/O ports.

3.2 ADDING EXTERNAL ROM

Expansion beyond the internal ROM capacity is implemented through bus I/O operations. The boundaries at which instruction fetches from external ROM occur are at address 1024 for the INS8048, and 2048 for the INS8049. External memory fetches are automatic once these boundaries are crossed. The INS8050 contains 4K of ROM, the maximum memory the 48-Series devices can directly address.

The expansion control signal ALE, is always present, $\overline{\text{PSEN}}$ is present only when reading from external memory.

The following sequence of events occurs for all instruction fetches from external ROM:

- The 12-bit address is output from the program counter. The least significant 8 bits on the bus, the most significant 4 bits on the lower half of Port 2.
- The address latch enable (ALE) signal goes high to indicate an address is output. The address is externally latched on the falling edge of ALE.
- The program store enable ($\overline{\text{PSEN}}$) signal goes low to enable the external ROM. The instruction output by ROM (onto the bus) is internally latched on the rising edge of $\overline{\text{PSEN}}$.
- The bus goes TRI-STATE and the processor operates on the instruction.

An illustration of instruction fetch timing is shown in *Figure 3-1*.

The external access (EA) pin, when driven high, causes all instruction fetches to be from external ROM. Internal ROM is disabled. For more detailed information, see 2.3.4.

Since the INS8035, INS8039, and INS8040 have no internal ROM, all instruction fetches are external (EA should always be high).

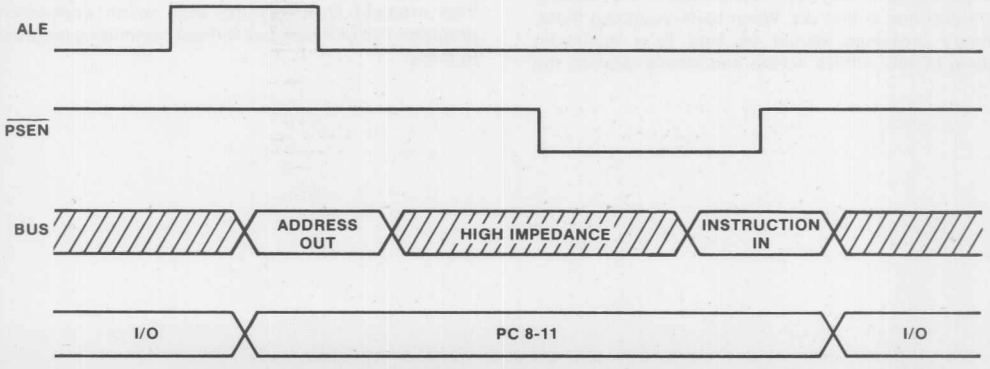


FIGURE 3-1. External Instruction Fetch Timing

87-16

3.2.1 Port 2 Restoration

Although the lower four bits of Port 2 output the four most significant address bits during the external instruction fetches, data is output during other portions of the machine cycle. If the information present on the quasi-bidirectional ports must remain stable, it can be latched on the rising edge of ALE. For additional information, see 2.3.13.

3.2.2 Bank-Switching

Programs less than 2K in length are addressed in the normal manner. Programs beyond the 2K addressing range can be accessed by using the bank-select instruction (SEL MB1) followed by a jump or call instruction. In this manner, the effective range of the branch instructions is extended beyond the normal 2K range, while preventing the user from inadvertently crossing the 2K boundary.

The bank-select instructions set a latch (DBF) to either a logic 1 or a logic 0, depending upon the bank desired. Once set, the latch remains set until another bank select instruction is executed. Bit 11 of the program counter is not incremented, as are the other bits of the program counter. Instead, the content of the DBF latch is loaded into bit 11 when a jump or call is executed. In this manner, depending upon the content of the DBF latch, instruction fetches will be from bank 0 (0 through 2047) or bank 1 (2048 through 4095). Since all twelve bits of the program counter are saved on the stack during subroutine calls, subroutines across the 2K boundary may be called. The proper bank will be restored upon return, but the DBF latch will not be altered. For additional information, see 2.2.5.

Direct memory addressing for ROM is limited to 4K by the 12-bit address, and only 256 locations of RAM are directly addressable by the 8-bit pointer registers R0 and R1. These limitations can be overcome by dedicating selected I/O lines for specific banks of memory, be it ROM or RAM. These lines can then be logically combined with the normal memory and I/O chip-select signals to enable the desired bank.

If I/O lines are used in this sort of address space expansion, the system software must maintain control over the bank-switching, so that each time a boundary is crossed, an I/O port line is first set. When bank-switching ROM, boundary crossings should be kept to a minimum. Jumping to subroutines across boundaries requires the programmer keep track of the return bank when the subroutine is completed. A software stack should be used to maintain the bank-switch bit. The stack can then be 'popped' upon completion of the subroutine to restore execution to the proper bank.

3.2.3 Interrupt Handling

Interrupts always vector the program counter to locations 3 or 7 in the lower 2K of ROM. Bit 11 of the program counter is forced to 0 and held there during the interrupt service routine. Therefore, interrupt service routines must reside solely in the lower 2K of ROM. Execution of a bank-select instruction within an interrupt service routine will not alter to bit 11 of the program counter, but will change the content of the DBF latch. The end of the interrupt service routine is signalled by the execution of the return and restore PSW (RETR) instruction.

For additional information on interrupts, see 2.3.3, 4.1.5, and 4.4.1.

3.2.4 Expansion Examples

A typical example of ROM expansion is illustrated in *Figure 3-2*.

In *Figure 3-2*, three external EPROMs have been added to an INS8048 for a total of 4K of ROM.

The Bus port initially outputs the lower eight external address bits. These bits are latched in an INS8212 by ALE for the external memory. The upper four address bits are output on the lower four lines of Port 2. Since these outputs are stable during the program memory fetch, they do not have to be latched. The two upper address bits (at P_{20}-P_{21}) connect directly to the upper two bits of external memory. The remaining two port output bits are decoded (and enabled by $\overline{PSEN}$) to generate the required chip select signals. The thus addressed data is output by the EPROMs onto the Bus and loaded internally during $\overline{PSEN}$.

Another method of adding external ROM is shown in *Figure 3-3*. For this example, a single 1K ROM has been added to an INS8040. No chip select decoding is necessary, $\overline{PSEN}$ selects the external ROM. All addressing is done as described for the preceeding example of 3-2. This example illustrates the ease which applications programs can be developed without committing programs to mask.

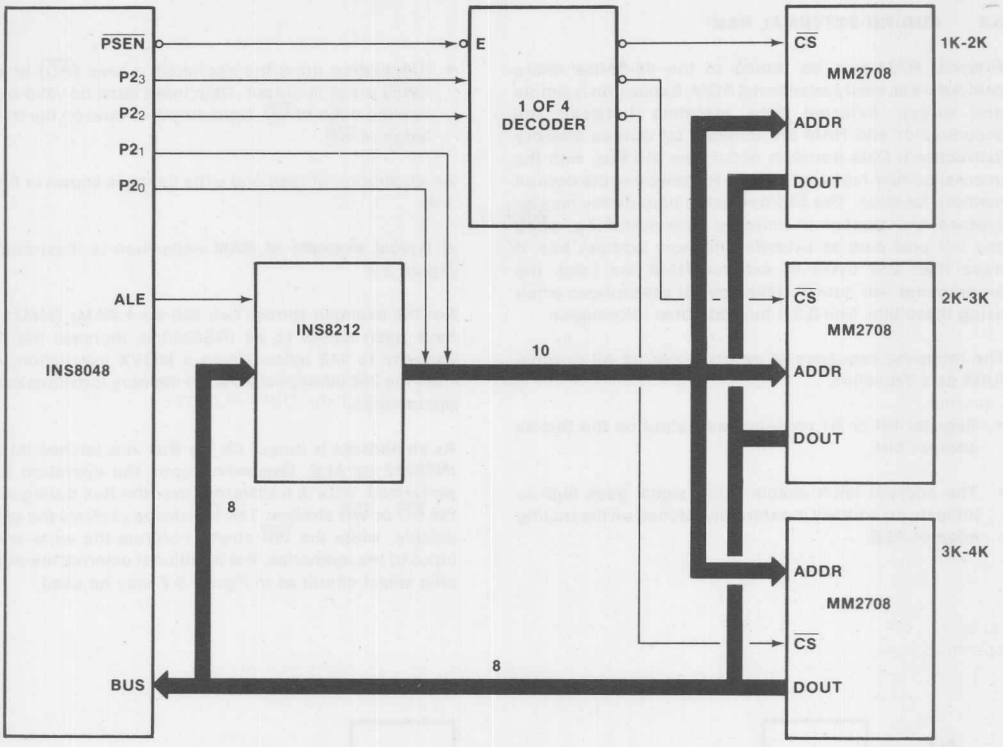


FIGURE 3-2. Three-Chip Memory Expansion

87-17

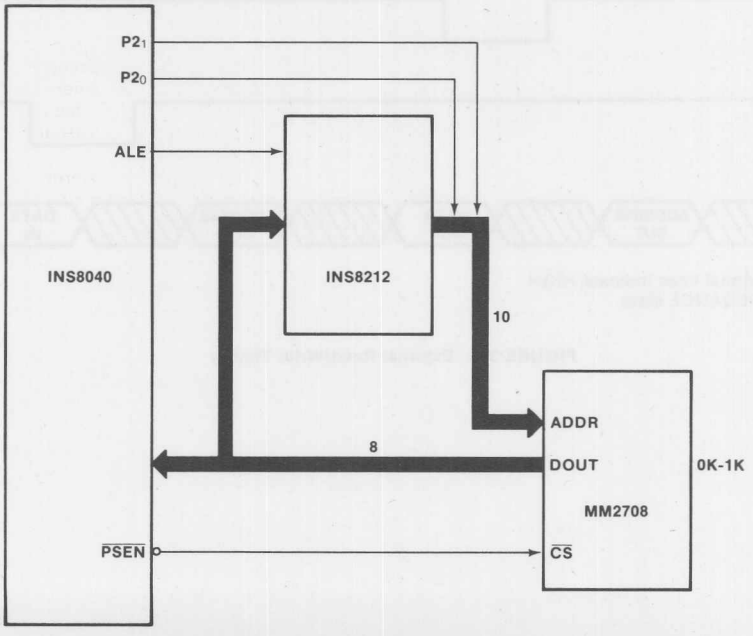


FIGURE 3-3. Single-Chip Memory Expansion

87-18

3.3 ADDING EXTERNAL RAM

External RAM can be added to the 48-Series microcomputers as easily as external ROM. Expansion is simple and direct. External data transfers between the accumulator and RAM are initiated by various memory instructions. Data transfers occur over the Bus, with the internal pointer registers, R0 and R1, selecting the desired memory location. The 256-byte page boundaries may be crossed any number of times by bank-switching, using any I/O port pins as extended memory address bits. If more than 256 bytes of external RAM are used, the programmer will have to take special precautions when using these bits. See 3.2.4 for additional information.

The following sequence of events occur for all external RAM data Transfers:

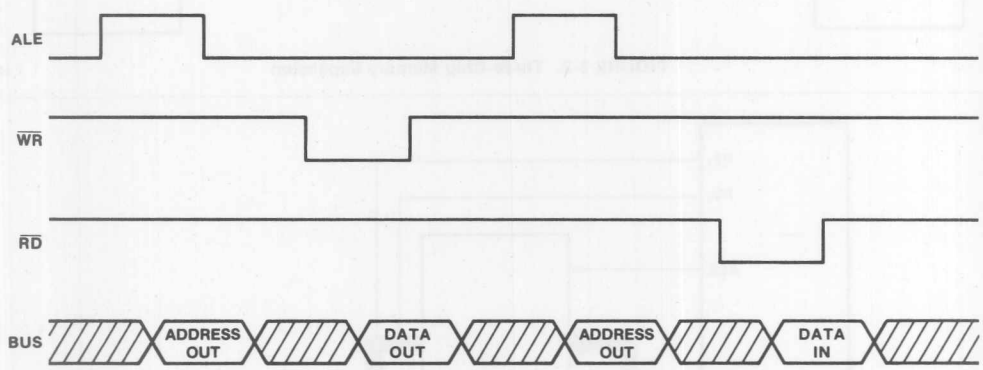
- Register R0 or R1 contents are output on the Bus as address bits.
- The address latch enable (ALE) signal goes high to indicate an address is externally latched on the trailing edge of ALE.
- Depending upon the operation, a read ($\overline{RD}$) or write ($\overline{WR}$) pulse is output. Data input must be valid on the trailing edge of $\overline{RD}$. Data output is valid on the trailing edge of $\overline{WR}$.

An illustration of read and write timing is shown in *Figure 3-4*.

A typical example of RAM expansion is illustrated in *Figure 3-5*.

For the example shown, two 256-by-4 RAMs (MM2111s) have been added to an INS8050 to increase the RAM capacity to 512 bytes. Using a MOVX instruction, data from the 256 additional external memory locations can be operated on.

As an address is output on the Bus, it is latched into the INS8212 by ALE. Depending upon the operation to be performed, data is transferred over the Bus during either the $\overline{RD}$ or $\overline{WR}$ strobes. The $\overline{RD}$ strobe controls the output disable, while the $\overline{WR}$ strobe controls the write enable input to the memories. For additional external memory, a chip select circuit as in *Figure 3-2* may be used.



NOTE: Diagonal lines indicate HIGH IMPEDANCE state.

FIGURE 3-4. External Read/Write Timing 87-19

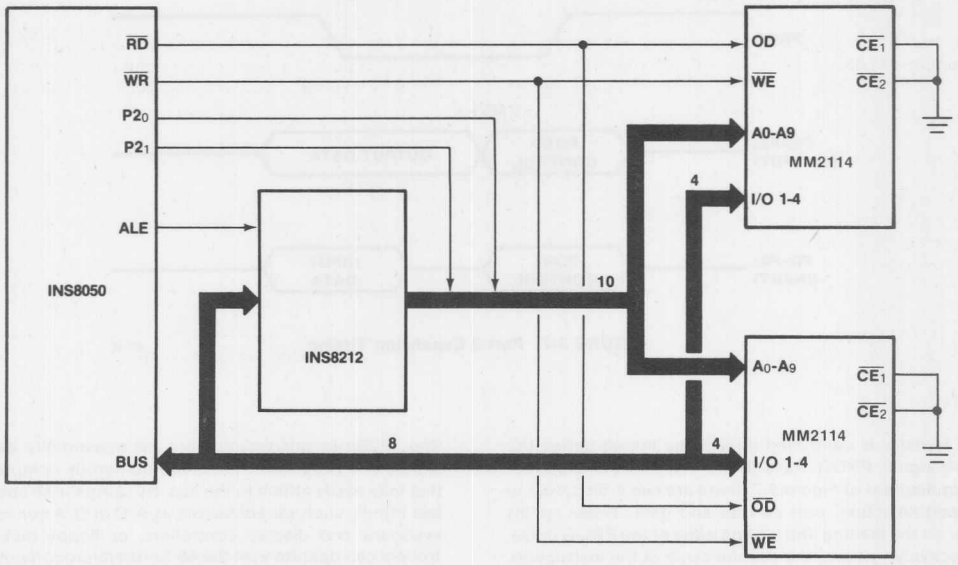


FIGURE 3-5. External RAM Expansion 87-20

3.4 INPUT/OUTPUT EXPANSION

The INS8243, an input/output port expansion device, increases the I/O capability of the 48-Series from 24 lines to 40 lines. The INS8243 connects directly to the lower four I/O lines of Port 2 and serves to expand those four I/O lines up to 16 lines. An illustration of the INS8243 connected to the INS8048 is shown in *Figure 3-6*. The added four ports have their own dedicated instructions and are addressed as ports 4 through 7. The lower three ports are the Bus (Port 0), I/O Port 1, and I/O Port 2.

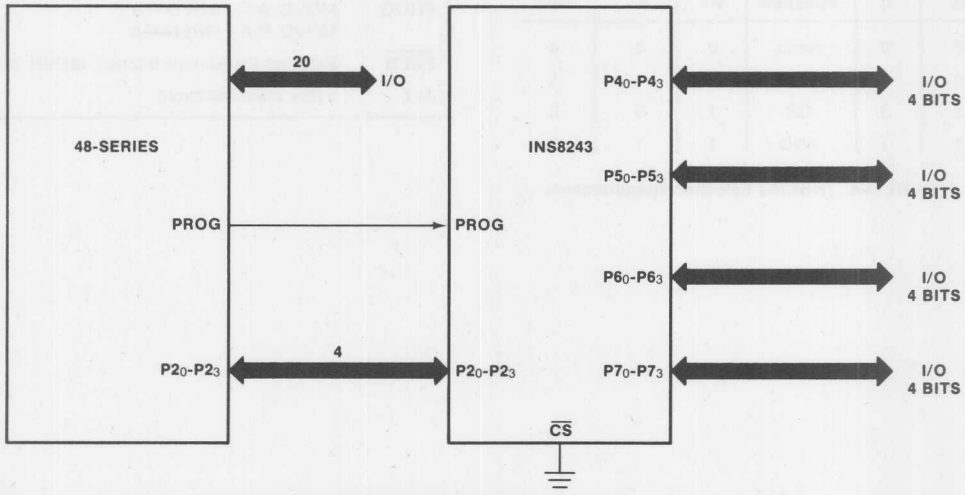


FIGURE 3-6. I/O Port Expansion 6-1

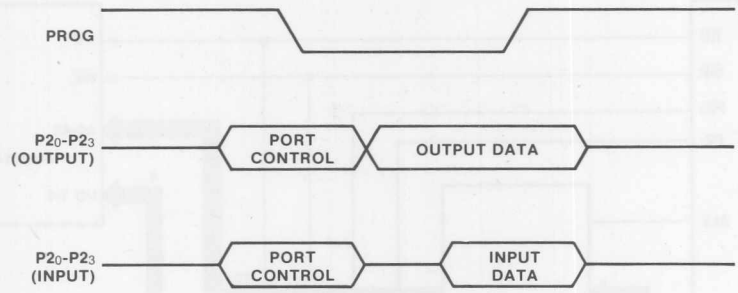


FIGURE 3-7. Port 2 Expansion Timing 87-21

Data transfer is controlled directly by the 48-Series I/O enable signal, PROG. As is shown in the output expander timing diagram of *Figure 3-7*, there are two 4-bit cycles to I/O port selection, port control and data. These cycles occur on the leading and trailing edge of the PROG pulse, respectively - during the second cycle of the instruction.

The first, or port control cycle is comprised of two 2-bit portions. Bit 0 and 1 select the port, while bits 3 and 4 determine the function. The second cycle is comprised of the 4-bit data. A detailed breakdown of INS8243 port selection is given in *Figure 3-8*.

Additional expansion ports can be added to the port expansion 'bus'. By using the upper four bits of Port 2 as chip select signals, four INS8243s can be added for a total of 64 I/O ports.

I_3	I_2	Function	P_1	P_0	Port
0	0	Read	0	0	4
0	1	Write	0	1	5
1	0	OR	1	0	6
1	1	AND	1	1	7

FIGURE 3-8. INS8243 Selection Requirements

The 48-Series microcomputers are expandable beyond simple I/O expansion. There are numerous components that may easily attach to the bus. By using the strobes and test inputs, such varied devices as A/D or D/A converters, keyboard and display controllers, or floppy disk controllers can operate with the 48-Series microcomputers. A list of 48-Series-compatible components is contained in Chapter 6.

A list of the various control signals used with I/O components is given in *Table 3-1*. The listed signals are inactive at all times other than for the given instructions.

TABLE 3-1. I/O Control Signals

Signal	Active During
$\overline{RD}$	MOVX A, @R or INS BUS
$\overline{WR}$	MOVX @R,A or OUTL BUS
PROG	MOVD A,P ; ANLD P,A ; MOVD P,A ; ORLD P,A
$\overline{PSEN}$	External instruction fetches; MOVP, MOVP3
ALE	Every machine cycle

Chapter 4

The 48-Series Instruction Set

4.1 INTRODUCTION

The assembly-language instruction set for the 48-Series microcomputers provides the following ten types of operations:

- Control
- Data Move
- Timer/Counter
- Accumulator
- Branch
- Input/Output
- Register
- Subroutine
- Flags
- Miscellaneous

The instruction statements, when assembled, generate object code. The object code, in turn, defines the precise operation the 48-Series microcomputers perform.

The 48-Series instruction set contains a total of 96 instructions designed for ease-of-use and to be memory efficient. The instructions are either one or two bytes in length, with over 70% of the instructions one byte in length. The double-byte instructions include all immediate instructions, certain I/O instructions (excluding MOV Pp,A), and most jump instructions.

Instruction execution occurs in either one or two machine cycles, with over 60% of the instructions executing in one machine cycle. Typical execution times, when using an 11 MHz crystal, are 1.36μsec and 2.72μsec, for one- and two-byte instructions, respectively.

The 48-Series microcomputers are efficient in handling both binary and BCD arithmetic operations. Additionally, the 48-Series can easily manipulate single-bits for control operations, as well as supply special instructions to handle loop counters, table-lookups and N-way branches.

4.1.1 Control Instructions

The control instructions allow the program to control interrupts, memory bank selection, and internal clock output.

Following initial power-on, the interrupt input ($\overline{INT}$) is automatically disabled. The external interrupt input can then be enabled or disabled using two of the control instructions. Additionally, $\overline{INT}$ is disabled while an interrupt is being processed and is re-enabled once the interrupt routine is completed.

The four bank-select instructions designate which bank of internal memory is accessed; two banks for ROM and two for RAM. A detailed explanation of the ROM bank-select is given in section 3.2.1. The working register bank-select instructions allow the programmer to substitute another bank of registers for the one presently in use. The bank-select instructions provide both an effective 16 working registers and a method for saving register contents on an interrupt or subroutine call. For the latter scheme, the user has the option of switching or not switching banks on interrupt. Bank-select status, as part of the program status word, is saved on the stack during subroutine calls. Restoration is optional, depending upon the return instruction used. If the switch-on interrupt is used, use of the "return and restore" status instruction to complete the interrupt service routine will automatically restore the originally selected bank.

The enable clock output instruction enables the internal clock to pin T0. Before this instruction is used, T0 is an input testable by the JT0 and JNT0 instructions. The internal clock frequency is the XTAL input frequency divided by three. The resultant output can then be used as a general purpose clock for the remainder of the system.

NOTE: *After the enable clock output instruction has been executed, the clock output at T0 is disabled only when $\overline{RESET}$ goes low.*

4.1.2 Data Move Instructions

The data move instructions are the primary instructions for moving data back and forth between the accumulator, various registers, or system memory. Data transfers from the registers 0-7 is direct (the instruction specifies the source or destination register). Data transfers from internal or external RAM is done indirectly via an address put in either registers R0 or R1 of the active bank. Transfers to or from internal RAM take only one machine cycle, whereas transfers to or from external RAM take two machine cycles. Data stored in the internal ROM can also be loaded directly into the accumulator. Additionally, data transfers can be made directly between the accumulator and either the internal timer/counter or the program status register. The ability to change the contents of the program status register provides an alternate means of restoring status after an interrupt, or altering the stack pointer, if necessary.

There is a Data Move instruction (XCHD A) that, working in conjunction with the SWAP A instruction, permits easy handling of 4-bit quantities, including BCD numbers. This instruction exchanges the lower 4 bits of the accumulator with the lower 4 bits of any internal RAM location. When used, with SWAP A, this instruction makes it easy to handle 4-bit values, including BCD numbers.

4.1.3 Timer/Counter Instructions

The timer/counter instructions enable the on-board 8-bit timer/counter, move data between the accumulator and the timer/counter, and start or stop the timer/counter.

The timer/counter can be used as either a crystal-controlled timer using the internal clock as its clock source, or an event counter (or timer) taking its input from the T1 input. Either application can be selected under direct software control. The start instruction used will determine which clock source is used.

The timer/counter can be loaded or read from the accumulator while the counter is either running or stopped. Regardless of the timer/counter operational mode, the stop count instruction will stop the timer/counter. Additionally, two instructions enable or disable the timer/counter interrupt flag for timer/counter output.

4.1.4 Accumulator Instructions

The accumulator instructions provide for adding data to (with or without carry), or for performing logical functions with, the accumulator. These operations of ADD, AND, OR, XOR, or rotates are performed upon immediate data, data memory, or the selected bank of registers. Data is moved between the accumulator and the various registers or memory depending upon the instruction used. A special instruction, SWAP A, allows the position of the two 4-bit nibbles in the accumulator to be swapped. This instruction is used in conjunction with the XCHD A instruction. See 4.1.2 for additional information.

A Decimal Adjust instruction has been included to facilitate BCD arithmetic. This instruction corrects the binary addition of two 2-digit BCD numbers in the accumulator to produce the correct BCD number.

Additional accumulator operations consist of rotates right and left (with or without carry), complement, increment, decrement, or clear. The rotate instructions are performed one bit at a time.

Although there is no subtract instruction in the 48-Series instruction set, subtraction can be performed using a simple three-byte set of instructions. The result of the subtraction will be retained in the accumulator. The instruction string is:

CPL A	Complement the accumulator
ADD A	Add the value to the accumulator
INC A	Add 1 to the accumulator

The contents of the program status word can be pulled into the accumulator for modification using the MOV A, PSW instruction. Once modified, the new contents can be placed into the PSW using MOV PSW, A.

Not only does this permit modification of the PSW flags, but the stack pointer contents may also be modified. By modifying the stack pointer, the contents of various locations within the stack may be used by Interrupts, or Call and Return instructions.

If a program is written that uses the contents of the stack pointer, a MOV A, PSW instruction must be used. Following the MOV A, PSW, the accumulator contents must be shifted left by one and two subtracted (or added, depending upon which next location is to be looked at) from the contents for the actual memory location within the stack.

Figure 2-3 and *Table 2-3* illustrate the stack and program status word, respectively.

4.1.5 Branch Instructions

The branch instructions allow jumps (unconditional and conditional) throughout memory.

The unconditional jump instruction permits jumps anywhere within the lower 2K of ROM. To jump to the upper 2K of ROM, a bank-select instruction must precede the jump instruction. Although the bank-select must precede the jump, the bank switch does not occur until the jump is executed. Once a bank is selected, all jumps will be within that bank until the bank-select/jump sequence is repeated.

The conditional jumps can test the following inputs or machine status. If the inputs or status conditions are true, the jump is permitted.

- T0 = 1 or 0
- T1 = 1 or 0
- $\overline{INT}$ = 0
- Accumulator = 0 or ≠ 0
- Accumulator bit = 1
- Carry = 1 or 0
- F0 = 1
- F1 = 1
- Timer Flag = 1

The conditional jumps allow branching to any address within the current page (256 words) under execution. The test conditions are the instantaneous values present when the conditional jump is executed.

The decrement register and skip if not zero instruction is useful for creating iteration counters. This instruction can designate any one of the eight registers in the currently selected bank to be a counter. The counter can effect a branch to any address within the current page.

The indirect jump instruction allows vectoring programs based upon the contents of the accumulator. The accumulator points to a location in the current page of ROM which contains the 8-bit jump address (also within the current page).

4.1.6 Input/Output Instructions

The input/output instructions provide for data transfers between the accumulator and the I/O ports. The I/O ports have latched outputs, while the inputs must be read when input data is valid.

Additional instructions permit the ANDing or ORing of immediate data from ROM to either the Bus, Port 1, or Port 2, with the result latched at the port. The capability to perform logical operations directly on the ports permits 'masks' stored in ROM to selectively set or reset individual bits of all three ports. Input on any given line via Ports 1 and 2 is enabled by first writing a logic 1 to each line where input is desired.

The third I/O port, the BUS port, is a true bi-directional port that can be either latched or treated as a fully synchronous, bidirectional bus. The BUS port can also have logical operations performed directly to its outputs. Unlike Ports 1 and 2, all eight BUS lines must be treated as either inputs or outputs at any given time. If the BUS is being used as an I/O port, it will be in the input mode from the time the device is reset until an OUTL BUS,A instruction is executed. From then on, the BUS will not be resettable as an input unless the programmer uses it for either an external bus access or the device itself is reset. The BUS port can operate synchronously with external RAM using the MOVX instructions. The I/O instructions generate either a read ($\overline{RD}$) or write ($\overline{WR}$) pulse, depending upon the operation. When either $\overline{RD}$ or $\overline{WR}$ is generated, data must be valid on the trailing edge of the pulse. When the BUS lines are not in use, they are in the TRI-STATE (high-impedance) mode.

The 48-Series I/O ports can be expanded from the basic three up to seven. Expansion is via the lower four bits of Port 2. Each of the expansion ports, 4 through 7, are 4-bit ports that have dedicated instruction types. The expansion port AND and OR instructions combine the accumulator contents with the selected port rather than with immediate data as on Ports 0 through 2.

The BUS latching instruction, OUTL BUS, is for use in single-chip operations where BUS is not used as an expansion port. If necessary, the OUTL BUS and MOVX instructions can be mixed. Care must be taken though, since data previously latched will be destroyed by executing MOVX. The BUS lines will then be left in their TRI-STATE (high-impedance) mode.

NOTE: *The OUTL instruction should never be used in systems with external ROM. If BUS is latched, the next instruction, if fetched from external ROM, may be fetched improperly.*

System expansion can also be done via the BUS port. For expanded systems, additional I/O ports can be memory-mapped, using the external RAM address space as addressed by pointer registers R0 or R1.

4.1.7 Register Instructions

The register instructions allow the programmer to increment or decrement any of the enabled internal registers. Additionally, the contents of a selected RAM location, as selected by the contents of R0 or R1, can be directly incremented.

4.1.8 Subroutine Instructions

The subroutine instructions are used to call or return from a designated subroutine.

Subroutines can be called from one bank to another as long as a bank-select instruction precedes a subroutine call. Once the subroutine in the other bank is completed, execution will return to the bank originally selected.

NOTE: *If the original bank is not reselected following a subroutine call to another bank, the next encountered jump instruction will transfer execution to the bank containing the subroutine.*

There are two return-from-subroutine instructions. One restores status to the upper four bits of the program status word, while the other does not. The return-and-restore-status instruction also signals the end of an interrupt service routine if one has been in progress.

4.1.9 Flag Instructions

The flag instructions allow the programmer to complement or clear three of the four user-accessable flags; Carry, F0, and F1. The functions of the four flags are as follows:

- Carry - Indicates an overflow occurred during a previous accumulator operation.
- Auxillary Carry - Indicates overflow between BCD digits when adding. Used by the decimal adjust instruction.
- F0 and F1 - General purpose flags used for conditional jump tests.

The carry flags and F0 are accessible as a part of the program status word and are stored on the stack during subroutine calls. Restoration is optional, dependent upon the return instruction used.

4.1.10 No Operation

This instruction does exactly what its name implies; it performs no operation other than take up memory space and execution time. Program execution continues with the next instruction. No Op is a useful tool during debug by saving 'space' to 'patch-in' additional instructions at some later time.

4.1.11 Instruction Set Summary

Table 4-1 is a summary of the instruction set for the 48-Series microcomputers.

Table 4-1. 48-Series Instruction Set Summary

MNEMONIC	FUNCTION	DESCRIPTION	CYCLES	BYTES	FLAGS C	AC	F0	F1
		ACCUMULATOR						
ADD A, #data	(A) ← (A) + data	Add Immediate the specified Data to the Accumulator.	2	2	•	•		
ADD A, Rr	(A) ← (A) + (Rr) for r = 0 - 7	Add contents of designated register to the Accumulator.	1	1	•	•		
ADD A, @ Rr	(A) ← (A) + ((Rr)) for r = 0 - 1	Add Indirect the contents the data memory location to the Accumulator.	1	1	•	•		
ADDC A, #data	(A) ← (A) (C) + data	Add Immediate with carry the specified data to the Accumulator.	2	2	•	•		
ADDC A, Rr	(A) ← (A) + (C) + (Rr) for r = 0 - 7	Add with carry the contents of the designated register to the Accumulator.	1	1	•	•		
ADDC A, @ Rr	(A) ← (A) + (C) + ((Rr)) for r = 0 - 1	Add Indirect with carry the contents of data memory location to the Accumulator.	1	1	•	•		
ANL A, #data	(A) ← (A) AND data	Logical AND specified Immediate Data with Accumulator.	2	2				
ANL A, Rr	(A) ← (A) AND (Rr) for r = 0 - 7	Logical AND contents of designated register with Accumulator.	1	1				
ANL A, @ Rr	(A) ← (A) AND ((Rr)) for r = 0 - 1	Logical AND Indirect the contents of data memory with Accumulator.	1	1				
CPL A	(A) ← NOT (A)	Complement the contents of the Accumulator.	1	1				
CLR A	(A) ← 0	CLEAR the contents of the Accumulator.	1	1				
DA A		DECIMAL ADJUST the contents of the Accumulator.	1	1	•			
DEC A	(A) ← (A) - 1	DECREMENT by 1 the accumulator's contents.	1	1				
INC A	(A) ← (A) + 1	Increment by 1 the accumulator's contents	1	1				
ORL A #data	(A) ← (A) OR data	Logical OR specified immediate data with Accumulator.	2	2				
ORL A, Rr	(A) ← (A) OR (Rr) for r = 0 - 7	Logical OR contents of designated register with Accumulator.	1	1				
ORL A, @ Rr	(A) ← (A) OR ((Rr)) for r = 0 - 1	Logical OR Indirect the contents of data memory location with Accumulator.	1	1				
RLA	(An + 1) ← (An) for n = 0 - 6 (A0) ← (A7)	Rotate Accumulator left by 1-bit without carry.	1	1				
RLC A	(An+1) ← (An); n = 0-6 (A0) ← (C) (C) ← (A7)	Rotate Accumulator left by 1-bit through carry.	1	1	•			
RR A	(An) ← (An+1); n = 0-6 (A7) ← (A0)	Rotate Accumulator right by 1-bit without without carry.	1	1				
RRC A	(An) ← (An+1); n + 0-6 (A7) ← (C) (C) ← (A0)	Rotate Accumulator right by 1-bit through carry.	1	1	•			
SWAP A	(A4-A7) ↔ (A0 - A3)	Swap the 2, 4-bit nibbles in the Accumulator.	1	1				
XRL A, #data	(A) ← (A) XOR data	Logical XOR immediate specified data with Accumulator.	2	2				
XRL A, Rr	(A) ← (A) XOR (Rr) for r = 0 - 7	Logical XOR contents of designated register with Accumulator.	1	1				
XRL A, @ Rr	(A) ← (A) XOR ((Rr)) for r = 0 - 1	Logical XOR Indirect the contents of data memory location with Accumulator.	1	1				

Table 4-1. 48-Series Instruction Set Summary (Cont'd.)

MNEMONIC	FUNCTION	DESCRIPTION	CYCLES	BYTES	FLAGS C	AC	F0	F1
		BRANCH						
DJNZ Rr, addr	(Rr) ← (Rr) -1; r = 0-7 if (Rr) ≠ 0; (PC 0-7) ← addr	Decrement the specified register and test contents.	2	2				
JBb addr	(PC 0-7) ← addr if Bb = 1 (PC) ← (PC) + 2 if Bb = 0	Jump to specified address if Accumulator bit is set.	2	2				
JC addr	(PC 0-7) ← addr if C = 1 (PC) ← (PC) +2 if C = 0	Jump to specified address if carry flag is set.	2	2				
JF0 addr	(PC 0-7) ← addr if F0 = 1 (PC) ← (PC) + 2 if F0 = 0	Jump to specified address if Flag F0 is set.	2	2				
JFI addr	(PC 0-7) ← addr if F1 = 1 (PC) ← (PC) + 2 if F1 = 0	Jump to specified address if Flag F1 is set.	2	2				
JMP addr	(PC 8-10) ← addr 8-10 (PC 0-7) ← addr 0-7 (PC 11) ← DBF	Direct Jump to specified address within the 2K address block.	2	2				
JMPP @ A	(PC 0-7) ← ((A))	Jump indirect to specified address pointed to by the accumulator in current page.	2	1				
JNC addr	(PC 0-7) ← addr if C = 0 (PC) ← (PC) + 2 if C = 1	Jump to specified address if carry flag is low.	2	2				
JNI addr	(PC 0-7) ← addr if I = 0 (PC) ← (PC) + 2 if I = 1	Jump to specified address if interrupt is low.	2	2				
JNT0 addr	(PC 0-7) ← addr if T0 = 0 (PC) ← (PC) + 2 if T0 = 1	Jump to specified address if Test 0 is low.	2	2				
JNT1 addr	(PC 0-7) ← addr if T1 = 0 (PC) ← (PC) + 2 if T1 = 1	Jump to specified address if Test 1 is low.	2	2				
JNZ addr	(PC 0-7) ← addr if A ≠ 0 (PC) ← (PC) + 2 if A = 0	Jump to specified address if accumulator is non-zero.	2	2				
JTF addr	(PC 0-7) ← addr if TF = 1 (PC) ← (PC) + 2 if TF = 0	jump to specified address if Timer Flag is set to 1.	2	2				
JT0 addr	(PC 0-7) ← addr if T0 = 1 (PC) ← (PC) + 2 if T0 = 0	Jump to specified address if Test 0 is a 1.	2	2				
JT1 addr	(PC 0-7) ← addr if T1 = 1 (PC) ← (PC) + 2 if T1 = 0	Jump to specified address if Test 1 is a 1.	2	2				
JZ addr	(PC 0-7) ← addr if A = 0 (PC) ← (PC) + 2 if A = 1	Jump to specified address if Accumulator is 0.	2	2				
		CONTROL						
EN I		Enable the External Interrupt input.	1	1				
DIS I		Disable the External Interrupt input.	1	1				
ENT0 CLK		Enable T0 as the Clock Output.	1	1				
SEL MB0	(DBF) ← 0	Select Bank 0 (locations 0 - 2047) of Program Memory.	1	1				
SEL MB1	(DBF) ← 1	Select Bank 1 (locations 2048 - 4095) of Program Memory.	1	1				
SEL RB0	(BS) ← 0	Select Bank 0 (locations 0 - 7) of Data Memory.	1	1				
SEL RB1	(BS) ← 1	Select Bank 1 (locations 24 - 31) of Data Memory.	1	1				
		DATA MOVES						
MOV A, #data	(A) ← data	Move Immediate the specified data into the Accumulator.	2	2				
MOV A, Rr	(A) ← (Rr); r = 0 - 7	Move the contents of the designated registers into the Accumulator.	1	1				
MOV A, @ Rr	(A) ← ((Rr)); r = 0 - 1	Move Indirect the contents of data memory location into the Accumulator.	1	1				

Table 4-1. 48-Series Instruction Set Summary (Cont'd.)

MNEMONIC	FUNCTION	DESCRIPTION	CYCLES	BYTES	FLAGS C	AC	F0	F1
		DATA MOVES (Cont'd.)						
MOV A, PSW	(A) ← (PSW)	Move contents of the Program Status Word into the Accumulator.	1	1				
MOV Rr, #data	(Rr) ← data; r = 0 - 7	Move Immediate the specified data into the designated register.	2	2				
MOV Rr, A	(Rr) ← (A); r = 0 - 7	Move Accumulator contents into the designated register.	1	1				
MOV @ Rr, A	((Rr)) ← (A); r = 0 - 1	Move Indirect Accumulator contents into data memory location.	1	1				
MOV @ Rr, #data	((Rr)) ← data; r = 0 - 1	Move Immediate the specified data into data memory.	2	2				
MOV PSW, A	(PSW) ← (A)	Move contents of Accumulator into the Program Status Word.	1	1	•	•	•	
MOVP A, @ A	(PC 0 - 7) ← (A) (A) ← ((PC))	Move the content of program memory location in the current page addressed by the content of accumulator into the accumulator.	2	1				
MOVP3 A, @ A	(PC 0 - 7) ← (A) (PC 8 - 10) ← 011 (A) ← ((PC))	Move the content of program memory location in page 3 address by the content of accumulator into the accumulator.	2	1				
MOVX A, @ R	(A) ← ((Rr)); r = 0 - 1	Move Indirect the contents of external data memory into the Accumulator.	2	1				
MOVX @ R, A	((Rr)) ← (A); r = 0 - 1	Move Indirect the contents of the Accumulator into external data memory.	2	1				
XCH A, Rr	(A) ↔ (Rr); r = 0 - 7	Exchange the Accumulator and designated register's contents.	1	1				
XCH A, @ Rr	(A) ↔ ((Rr)); r = 0 - 1	Exchange Indirect contents of Accumulator and location in data memory.	1	1				
XCHD A, @ Rr	(A0 - A3) ↔ (((Rr)) 0 - 3); r = 0 - 1	Exchange Indirect 4-bit contents of Accumulator and data memory.	1	1				
		TIMER COUNTER						
EN TCNTI		Enable Internal Interrupt Flag for Timer/Counter output.	1	1				
DIS TCNTI		Disable Internal Interrupt Flag for Timer/Counter output.	1	1				
MOV A, T	(A) ← (T)	Move contents of Timer/Counter into Accumulator.	1	1				
MOV T, A	(T) ← (A)	Move contents of Accumulator into Timer/Counter.	1	1				
STOP TCNT		Stop Count for Event Counter.	1	1				
STRT CNT		Start Count for Event Counter.	1	1				
STRT T		Start Count for Timer.	1	1				
		INPUT/OUTPUT						
ANL BUS, data	(BUS) ← (BUS) AND data	Logical AND Immediate specified data with contents of BUS.	2	2				
ANL Pp, data	(Pp) ← (Pp) AND data; p = 1 - 2	Logical AND immediate specified data with designated port (1 or 2).	2	2				
ANLD Pp, A	(Pp) ←(Pp) AND (A0 - A3); p = 4-7	Logical AND contents of Accumulator with designated port (4 - 7).	2	1				
IN A, Pp	(A) ← (Pp); p = 1-2	Input data from designated port (1 - 2) into Accumulator.	2	1				
INS A, BUS	(A) ← (BUS)	Input strobed BUS data into Accumulator	2	1				

Table 4-1. 48-Series Instruction Set Summary (Cont'd.)

MNEMONIC	FUNCTION	DESCRIPTION	CYCLES	BYTES	FLAGS C	AC	F0	F1
		INPUT/OUTPUT (Cont'd.)						
MOVD A, Pp	(A0-A3) ← (Pp); p = 4-7 (A4-A7) ← 0	Move contents of designated port (4 - 7) into Accumulator.	2	1				
MOVD Pp, A	(Pp) ← (A0 - A3); p = 4 - 7	Move contents of Accumulator to designated port (4 - 7).	2	1				
ORL BUS, #data	(BUS) ← (BUS) OR data	Logical OR Immediate specified data with contents of BUS.	2	2				
ORLD Pp, A	(Pp) ← (Pp) OR (A0 - A3); p = 4-7.	Logical OR contents of Accumulator with designated port (4 - 7).	2	1				
ORL Pp, #data	(Pp) ← (Pp) OR data; p = 1 - 2.	Logical OR Immediate specified data with designated port (1 - 2).	2	2				
OUTL BUS, A	(BUS) ← (A)	Output contents of Accumulator onto BUS.	2	1				
OUTL Pp, A	(Pp) ← (A); p = 1 - 2	Output contents of Accumulator to designated port (1 - 2).	1	1				
		REGISTERS						
DEC Rr	(Rr) ← (Rr) -1; r = 0-7	Decrement by 1 contents of designated register.	1	1				
INC Rr	(Rr) ← (Rr) +1; r = 0-7	Increment by 1 contents of designated register.	1	1				
INC @ Rr	((Rr)) ← ((Rr)) + 1; r = 0-1.	Increment Indirect by 1 the contents of data memory location.	1	1				
		SUBROUTINE						
CALL addr	((SP)) ← (PC) ((SP)) ←(PSW 4-7) (SP) ← (SP) + 1 (PC8-10) ← addr 8-10 (PC 0-7)← addr 0-7 (PC 11) ← DBF	Call designated Subroutine.	2	2				
RET	(SP) ← (SP) - 1 (PC) ← ((SP))	Return from Subroutine without restoring Program Status Word.	2	1				
RETR	(SP) ← (SP) - 1 (PC) ← ((SP)) (PSW 4-7) ← ((SP))	Return from Subroutine restoring Program Status Word.	2	1	•	•		
		FLAGS						
CPL C	(C) ← NOT (C)	Complement Content of carry bit.	1	1	•			
CPL F0	(F0) ← NOT (F0)	Complement Content of Flag F0.	1	1			•	
CPL F1	(F1) ← NOT (F1)	Complement Content of Flag F1.	1	1				•
CLR C	(C) ← 0	Clear content of carry bit to 0.	1	1	•			
CLR F0	(F0) ← 0	Clear content of Flag 0 to 0.	1	1			•	
CLR F1	(F1) ← 0	Clear content of Flag 1 to 0.	1	1				•
		MISCELLANEOUS						
NOP		No operation	1	1				

4.2 48-SERIES INSTRUCTION SET

The following pages contain detailed information on the 48-Series instruction set. The instructions have been arranged alphabetically, so they may be easily located. An illustration of the instruction set presentation is shown in *Figure 4-1*. A summary of the instruction set is provided in the INS8048-Series Data Sheet located in Appendix A. Special symbols and notations used with the instructions are also in the Data Sheet and listed in *Table 4-2*.

Mnemonic { MOV A,T

Operation { Move timer/counter to accumulator

Op Code { 0 1 0 0 | 0 0 1 0

Symbolic Representation { (A) ← (T)

Description { The contents of the timer/counter are moved into the accumulator

Special Conditions { Cycles: 1
Bytes: 2

FIGURE 4-1. 48-Series Sample Instruction

Table 4-2. Symbols Used in 48-Series Instructions

SYMBOL	DESCRIPTION
A	The Accumulator
AC	The Auxiliary Carry Flag
addr	Program Memory Address (12 bits)
Bb	Bit Designator (b = 0-7)
BS	The Bank Switch
BUS	The BUS Port
C	Carry Flag
CLK	Clock Signal
CNT	Event Counter
D	Nibble Designator (4 bits)
data	Number or Expression (8 bits)
DBF	Memory Bank Flip-Flop
F_0, F_1	Flags 0, 1
I	Interrupt
P	"In-Page" Operation Designator
P_p	Port Designator (p = 1, 2 or 4-7)
PSW	Program Status Word
Rr	Register Designator (r = 0, 1 or 0-7)
SP	Stack Pointer
T	Timer
TF	Timer Flag

TABLE 4-2. Symbols Used in 48-Series Instructions (Cont'd.)

SYMBOL	DESCRIPTION
T_0, T_1	Testable Flags 0, 1
X	External RAM
=	Prefix for Immediate Data
@	Prefix for Indirect Address
S	Program Counter's Current Value
(x)	Contents of External RAM Location
((x))	Contents of Memory Location Addressed by the Contents of External RAM Location
←	Replaced By

ADD A,Rr

Add contents of designated register to accumulator.

0	1	1	0	1	r	r	r

(A) ← (A) + (Rr); where r = 0 through 7

The contents of the internal register designated by bits 'r', are added to the accumulator.

Cycles: 1
Bytes: 1
Flags: Carry, Auxillary Carry

ADD A,@Rr

Add-indirect contents of RAM location to accumulator.

0	1	1	0	0	0	0	r

(A ← (A) + ((Rr)); where r = 0 or 1

The contents of the internal RAM location, as addressed by bits 0 through 5* of register 'r', are added to the accumulator.

*bits 0 through 6 for INS8039/INS8049
*bits 0 through 7 for INS8050
Cycles: 1
Bytes: 1
Flags: Carry, Auxillary Carry

ADD A,# data

Add-immediate specified data to accumulator.

0	0	0	0	0	0	1	1	byte 1
d7	d6	d5	d4	d3	d2	d1	d0	byte 2

(A) ← (A) + data

The data contained in byte 2 is added to the data in the accumulator.

Cycles: 2
Bytes: 2
Flags: Carry, Auxillary Carry

ADDC A,Rr

Add-with-carry contents of designated register to accumulator.

0	1	1	1	1	r	r	r

(A) ← (A) + (C) + (Rr); where r = 0 through 7

The content of the carry bit is added to accumulator location 0 as the contents of the register specified by the 'r' bits are added to the accumulator.

Cycles: 1
Bytes: 1
Flags: Carry, Auxillary Carry

ADDC A,@Rr

Add-indirect-with-carry contents of RAM location to accumulator.

0	1	1	1	0	0	0	r

(A) ← (A) + (C) + ((Rr)); where r = 0 or 1

The content of the carry bit is added to accumulator location 0 while the contents of the internal RAM location, as addressed by bits 0 through 5* of register 'r', are added to the accumulator.

*bits 0 through 6 for INS8039/INS8049
*bits 0 through 7 for INS8050
Cycles: 1
Bytes: 1
Flags: Carry, Auxillary Carry

ADDC A,# data

Add-with-carry specified immediate data to accumulator.

0	0	0	1	0	0	1	1	byte 1
d7	d6	d5	d4	d3	d2	d1	d0	byte 2

(A) ← (A) + (C) + data

The content of the carry bit is added to the accumulator location 0 as the data contained in byte 2 is added to the data in the accumulator.

Cycles: 2
Bytes: 2
Flags: Carry, Auxillary Carry

ANL A,Rr

Logical-AND contents of designated register with accumulator.

0	1	0	1	1	r	r	r

(A) ← (A) AND (Rr); where r = 0 through 7

The contents of the register specified by the 'r' bits are logically ANDed with the data in the accumulator.

Cycles: 1
Bytes: 1

ANL A,@Rr

Logical-AND-indirect contents of RAM with accumulator.

0	1	0	1	0	0	0	r

(A) ← (A) AND ((Rr)); where r = 0 or 1

The contents of the internal RAM location, as addressed by bits 0 through 5* of register 'r', are logically ANDed with the data in the accumulator.

*bits 0 through 6 for INS8039/INS8049
*bits 0 through 7 for INS8050
Cycles: 1
Bytes: 1

ANL A,# data

Logical-AND-immediate specified data with accumulator.

0	1	0	1	0	0	1	1	byte 1
d7	d6	d5	d4	d3	d2	d1	d0	byte 2

(A) ← (A) AND data

The data contained in byte 2 are logically ANDed with the data in the accumulator and the results are sent back to that port. Accumulator contents are not affected.

Cycles: 2
Bytes: 2

ANL BUS, # data

Logical-AND-immediate specified data with contents of BUS.

1	0	0	1	1	0	0	0	byte 1
d7	d6	d5	d4	d3	d2	d1	d0	byte 2

(BUS) ← (BUS) AND data

The data contained in byte 2 are logically ANDed immediately with the data on the BUS port and the results are sent back to that port. Use of this instruction asumes prior execution of an OUTL BUS A instruction.

Cycles: 2
Bytes: 2

ANL Pp, # data

Logical-AND-immediate specified data with designated port (1 or 2)

1	0	0	1	1	0	p1	p0	byte 1
d7	d6	d5	d4	d3	d2	d1	d0	byte 2

(Pp) ← (Pp) AND data; where p = 1 or 2

The data contained in byte 2 are logically ANDed immediately with the data on the port designated by bits 'p' and the results are sent back to that port. Accumulator contents are not affected. Op code bits 'p' designate the following ports:

Port	P1	P0
1	0	1
2	1	0

Cycles: 2
Bytes: 2

ANLD Pp, A

Logical-AND contents of accumulator with designated expansion port (4 through 7).

1	0	0	1	1	1	p1	p0

(Pp) ← (Pp) AND (A0 - 3); where p = 4 through 7

The data in accumulator, bits 0 through 3, are logically ANDed with the 4-bit data on the expander port designated by bits 'p' and the results are sent back to that port. Accumulator contents are not affected. Op code bits 'p' designate the following ports:

Port	p1	p0
4	0	0
5	0	1
6	1	0
7	1	1

Cycles: 2
Bytes: 1

CALL addr

Call designated subroutine.

a10	a9	a8	1	0	1	0	0	byte 1
a7	a6	a5	a4	a3	a2	a1	a0	byte 2

((SP)) ← (PC), (PSW 4-7)
(SP) ← (SP) + 1
(PC 8-10) ← addr 8-10
(PC 0-7) ← addr 0-7
(PC 11) ← DBF

The contents of both the program counter and program status word, bits 4 through 7, are saved on the stack. The stack pointer is incremented by one. The contents of the program counter are replaced by address bits 'a' from bytes 1 and 2. Address bit 11 in the program counter is determined by the most recent bank select instruction (SEL MB) executed.

NOTE: *Although the stack pointer is only incremented by one, internally it is incremented by two so the PSW and PC can be saved on the stack*

Upon return from the subroutine, program execution continues with the instruction immediately following CALL.

Cycles: 2
Bytes: 2

CLR A

Clear contents of accumulator to zero.

0	0	1	0	0	1	1	1

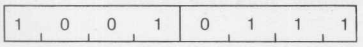
(A) ← 0

The contents of the accumulator are cleared to zero.

Cycles: 1
Bytes: 1

CLR C

Clear content of carry bit to zero.



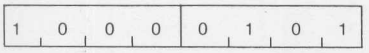
(C) ← 0

The content of the carry bit is cleared to zero. During normal program execution, the carry bit may be set to one due to an ADD, ADD C, RLCA, CPLC, RRCA or DAA instruction.

Cycles: 1
Bytes: 1
Flags: Carry

CLR F0

Clear content of flag 0 to zero.



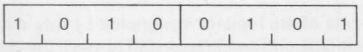
(F0) ← 0

The content of flag 0 is cleared to zero.

Cycles: 1
Bytes: 1
Flags: F0

CLR F1

Clear content of flag 1 to zero.



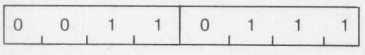
(F1) ← 0

The content of flag 1 is cleared to zero.

Cycles: 1
Bytes: 1
Flags: F1

CPL A

Complement contents of the accumulator.



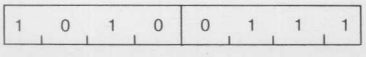
(A) ← NOT (A)

The contents of the accumulator are complemented. This is a one complement, with each 1 changing to a 0 and each 0 changing to a 1.

Cycles: 1
Bytes: 1

CPL C

Complement content of carry bit.



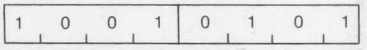
(C) ← NOT (C)

The content of carry bit is complemented. A content of 1 is changed to 0; a content of 0 is changed to 1.

Cycles: 1
Bytes: 1
Flags: Carry

CPL F0

Complement content of flag 0.



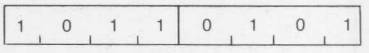
(F0) ← NOT (F0)

The content of flag 0 is complemented. A content of 1 is changed to 0; a content of 0 is changed to 1.

Cycles: 1
Bytes: 1
Flags: F0

CPL F1

Complement content of flag 1.



(F1) ← NOT (F1)

The content of flag 1 is complemented. A content of 1 is changed to 0; a content of 0 is changed to 1.

Cycles: 1
Bytes: 1
Flags: F1

DA A

Decimal-adjust contents of accumulator.

0	1	0	1	0	1	1	1

The 8-bit contents of the accumulator are adjusted to form two 4-bit BCD digits. Carry is affected. If accumulator bits 0 through 3 are greater than nine, or if the auxillary carry bit is a one, the accumulator is incremented by six. Accumulator bits 4 through 7 are then checked. If they exceed nine, the upper four bits are incremented by six. If an overflow occurs, carry is set to one.

NOTE: *AC is set any time the four LSB's of AC >9.*

Example: Assume accumulator contains
1 0 1 1 1 0 1 0

AC	C	7	6	5	4	3	2	1	0	
1	0	1	0	1	1	1	0	1	0	b0 - b3 > 9
						0	1	1	0	Add 6
0	0	1	1	0	0	0	0	0	0	b4 - b7 > 9
		0	1		0					Add 6
0	1	0	0	1	0	0	0	0	1	

Cycles: 1
Bytes: 1
Flags: Carry

DEC A

Decrement contents of accumulator by one.

0	0	0	0	0	1	1	1

(A) ← (A) - 1

The contents of the accumulator are decremented by one.

Cycles: 1
Bytes: 1

DEC Rr

Decrement contents of register Rr by one.

1	1	0	0	1	r	r	r

(Rr) ← (Rr) - 1; where r = 0 through 7

The contents of the register designated by bits 'r' are decremented by one.

Cycles: 1
Bytes: 1

DIS I

Disable external interrupt input.

0	0	0	1	0	1	0	1

The external interrupt input is disabled. Low-going signals at the interrupt input have no effect.

Cycles: 1
Bytes: 1

DIS TCNTI

Disable internal timer/counter interrupt flag.

0	0	1	1	0	1	0	1

The internal timer/counter interrupt flag output is disabled. A pending timer/counter interrupt request is cleared. If the timer is operating and overflows, the timer flag will be set, but no interrupt will occur, and the timer will continue to count.

Cycles: 1
Bytes: 1

DJNZ Rr, addr

Decrement specified register, test contents, jump if not zero.

1	1	1	0	1	r	r	r	byte 1
a7	a6	a5	a4	a3	a2	a1	a0	byte 2

(Rr) ← (Rr) - 1; where r = 0 through 7
(PC 0-7) ← addr, if (Rr) ≠ 0

The contents of the register designated by bits 'r' are decremented by one and then tested to see if the contents equal zero. If the register contents equal zero, the next sequential instruction is executed. If the register contents do not equal zero, control passes to the instruction at the address designated in byte 2.

The address is eight bits in length, limiting jumps to within the current 256-location page. If byte 1 of DJNZ is at location 255 of page 1 and byte 2 is at location 0 of page 2, the jump will be to the specified address within page 2.

Cycles: 2
Bytes: 2

EN I

Enable external interrupt.

0	0	0	0	0	1	0	1

The external interrupt input is enabled. A low-going signal at the interrupt input initiates a vector to location 3 in ROM. If the interrupt input is already low, this instruction will execute a call to location 3. See RETR instruction for additional information.

Cycles: 1
Bytes: 1

EN TCNTI

Enable internal timer/counter interrupt.

0	0	1	0	0	1	0	1

The internal timer/counter interrupt is enabled. A timer/counter overflow will set the timer flag and an interrupt vector to location 7 in ROM will occur. If the timer flag is already set indicating a timer/counter overflow, enabling the timer/counter flag output will not cause an interrupt. See RETR instructions for additional information.

Cycles: 1
Bytes: 1

ENT0 CLK

Enable clock output at T0.

0	1	1	1	0	1	0	1

Test input T0 is enabled to output the internal system clock. T0 is disabled as a test input. T0 is disabled as a clock output only by a system reset.

Cycles: 1
Bytes: 1

IN A, Pp

Input data to accumulator from designated port (1 or 2).

0	0	0	0	1	0	p1	p0

(A) ← (Pp); where p = 1 or 2

Data present at the port designated by bits 'p' is input into the accumulator. Opcode bits 'p' designate the following ports:

Port	p1	p0
1	0	1
2	1	0

Cycles: 1
Bytes: 1

INC A

Increment contents of accumulator by one.

0	0	0	1	0	1	1	1

(A) ← (A) + 1

The contents of the accumulator are incremented by one.

Cycles: 1
Bytes: 1

INC Rr

Increment contents of designated register by one.

0	0	0	1	1	r	r	r

(Rr) ← (Rr) + 1; where r = 0 through 7

The contents of the register designated by bits 'r' are incremented by one.

Cycles: 1
Bytes: 1

INC @ Rr

Increment-indirect contents of RAM by one.

0	0	0	1	0	0	0	r

((Rr)) ← ((Rr)) + 1; where r = 0 or 1

The contents of the internal RAM location, as addressed by bits 0 through 5* of register 'r', are incremented by one.

*bits 0 through 6 for INS8039/INS8049
*bits 0 through 7 for INS8040/INS8050
Cycles: 1
Bytes: 1

INS A, BUS

Input strobed data to accumulator from BUS.

0	0	0	0	1	0	0	0

(A) ← (BUS)

Data present at the Bus port is input to the accumulator during the $\overline{RD}$ strobe (see *Figure 2-8*).

Cycles: 2
Bytes: 1

JBb addr

Jump to specified address if accumulator bit 'b' is set.

b2	b1	b0	1	0	0	1	0	byte 1
a7	a6	a5	a4	a3	a2	a1	a0	byte 2

(PC 0-7) ← addr, if Bb = 1
(PC) ← (PC) + 2, if Bb = 0

If the accumulator bit designated by bits 'b' is set to a logic one, the contents of the program counter are replaced by address bits 'a' from byte 2. If bit 'b' in the accumulator is a logic zero, the next sequential instruction is executed. Bits b2, b1, and b0 represent a number from 0 to 7 designating which bit in the accumulator is to be tested.

Cycles: 2
Bytes: 2

JC addr

Jump to specified address if carry flag is set.

1	1	1	1	0	1	1	0	byte 1
a7	a6	a5	a4	a3	a2	a1	a0	byte 2

(PC 0-7) ← addr, if C = 1
(PC) ← (PC) + 2, if C = 0

If the carry bit is set to a logic one, the contents of the program counter are replaced by address bits 'a' from byte 2. If carry is a logic zero, the next sequential instruction is executed.

Cycles: 2
Bytes: 2

JF0 addr

Jump to specified address if flag 0 is set.

1	0	1	1	0	1	1	0	byte 1
a7	a6	a5	a4	a3	a2	a1	a0	byte 2

(PC 0-7) ← addr, if F0 = 1
(PC) ← (PC) + 2, if F0 = 0

If flag 0 is set to a logic one, the contents of the program counter are replaced by address bits 'a' from byte 2. If flag 0 is a logic zero, the next sequential instruction is executed.

Cycles: 2
Bytes: 2

JF1 addr

Jump to specified address if flag 1 is set.

0	1	1	1	0	1	1	0	byte 1
a7	a6	a5	a4	a3	a2	a1	a0	byte 2

(PC 0-7) ← addr, if F1 = 1
(PC) ← (PC) + 2, if F1 = 0

If flag 1 is set to a logic one, the contents of the program counter are replaced by address bits 'a' from byte 2. If flag 1 is a logic zero, the next sequential instruction is executed.

Cycles: 2
Bytes: 2

JMP addr

Jump-direct to specified address within 2K address block.

a10	a9	a8	0	0	1	0	0	byte 1
a7	a6	a5	a4	a3	a2	a1	a0	byte 2

(PC 8-10) ← addr 8-10
(PC 0-7) ← addr 0-7
(PC11) ← DBF

The contents of the program counter are replaced by address bits 'a' in bytes 1 and 2. Address bit 11 in the program counter is determined by the most recent bank select instruction (SEL MB) executed.

Cycles: 2
Bytes: 2

JMPP @ A

Jump-indirect to specified address within address page.

1	0	1	1	0	0	1	1	

(PC 0-7) ← ((A))

The contents of the program counter are replaced by the ROM contents within the current page pointed to by the accumulator. For example, if the accumulator contains X'20, a jump to the address stored at location 32 (in the current page) occurs.

Cycles: 2
Bytes: 1

JNC addr

Jump to specified address if carry flag is a logic zero.

1	1	1	0	0	1	1	0	bytes 1
a7	a6	a5	a4	a3	a2	a1	a0	byte 2

(PC 0-7) ← addr, if C = 0
(PC) ← (PC) + 2, if C = 1

If the carry flag is low, the contents of the program counter are replaced by address bits 'a' from byte 2. If carry is a logic one, the next sequential instruction is executed.

Cycles: 2
Bytes: 2

JNI addr

Jump to specified address if interrupt is a logic zero.

1	0	0	0	0	1	1	0	byte 1
a7	a6	a5	a4	a3	a2	a1	a0	byte 2

(PC 0-7) ← addr, if I = 0
(PC) ← (PC) + 2, if I = 1

If the interrupt input is at a logic zero, the contents of the program counter are replaced by address bits 'a' from byte 2. If the interrupt input is a logic one, the next sequential instruction is executed. This instruction provides a means for testing the condition of the external interrupt pin while it is disabled as an interrupt.

Cycles: 2
Bytes: 2

JNT0 addr

Jump to specified address if test 0 is a logic zero.

0	0	1	0	0	1	1	0	byte 1
a7	a6	a5	a4	a3	a2	a1	a0	byte 2

(PC 0-7) ← addr, if T0 = 0
(PC) ← (PC) + 2, if T0 = 1

If input T0 is at a logic zero, the contents of the program counter are replaced by address bits 'a' from byte 2. If T0 is a logic one, the next sequential instruction is executed. This instruction should not be executed after an ENT0 CLK instruction.

Cycles: 2
Bytes: 2

JNT1 addr

Jump to specified address if test 1 is a logic zero.

0	1	0	0	0	1	1	0	byte 1
a7	a6	a5	a4	a3	a2	a1	a0	byte 2

(PC 0-7) ← addr, if T1 = 0
(PC) ← (PC) + 2, if T1 = 1

If input T1 is at a logic zero, the contents of the program counter are replaced by address bits 'a' from byte 2. If T1 is a logic one, the next sequential instruction is executed.

Cycles: 2
Bytes: 2

JNZ addr

Jump to specified address if accumulator is non-zero.

1	0	0	1	0	1	1	0	byte 1
a7	a6	a5	a4	a3	a2	a1	a0	byte 2

(PC 0-7) ← addr, if A ≠ 0
(PC) ← (PC) + 2, if A = 0

If the contents of the accumulator are non-zero, the contents of the program counter are replaced by address bits 'a' from byte 2. If the accumulator contents are zero, the next sequential instruction is executed.

Cycles: 2
Bytes: 2

JTF addr

Jump to specified address if timer flag is set.

0	0	0	1	0	1	1	0	byte 1
a7	a6	a5	a4	a3	a2	a1	a0	byte 2

(PC 0-7) ← addr, if TF = 1
(PC) ← (PC) + 2, if TF = 0

If the internal timer/counter flag is set to a logic one, the contents of the program counter are replaced by address bits 'a' from byte 2. If the timer/counter flag is a logic zero, the next sequential instruction is executed.

Testing the timer/counter flag resets the flag to zero. This instruction provides a means of testing the timer/counter flag if the timer/counter interrupt is disabled. An overflow of the timer/counter will cause an interrupt vector to location 7 in ROM unless the timer/counter interrupt has be disabled.

Cycles: 2
Bytes: 2

JT0 addr

Jump to specified address if test 0 is a logic one.

0	0	1	1	0	1	1	0	byte 1
a7	a6	a5	a4	a3	a2	a1	a0	byte 2

(PC 0-7) ← addr, if T0 = 1
(PC) ← (PC) + 2, if T0 = 0

If input T0 is at a logic one, the contents of the program counter are replaced by address bits 'a' from byte 2. If T0 is a logic zero, the next sequential instruction is executed.

Cycles: 2
Bytes: 2

JT1 addr

Jump to specified address if test 1 is a logic one.

0	1	0	1	0	1	1	0	byte 1
a7	a6	a5	a4	a3	a2	a1	a0	byte 2

(PC 0-7) ← addr, if T1 = 1
(PC) ← (PC) + 2, if T1 = 0

If input T1 is at a logic one, the contents of the program counter are replaced by address bits 'a' from byte 2. If T1 is a logic zero, the next sequential instruction is executed.

Cycles: 2
Bytes: 2

JZ addr

Jump to specified address if accumulator is zero.

1	1	0	0	0	1	1	0	byte 1
a7	a6	a5	a4	a3	a2	a1	a0	byte 2

(PC 0-7) ← addr, if A = 0
(PC) ← (PC) + 2, if A ≠ 0

If the contents of the accumulator are zero, the contents of the program counter are replaced by address bits 'a' from byte 2. If the accumulator contents are non-zero, the next sequential instruction is executed.

Cycles: 2
Bytes: 2

MOV A, # data

Move-immediate specified data into accumulator.

0	0	1	0	0	0	1	1	byte 1
d7	d6	d5	d4	d3	d2	d1	d0	byte 2

(A) ← data

The data contained in byte 2 is moved into the accumulator.

Cycles: 2
Bytes: 2

MOV A, PSW

Move contents of program status word into accumulator.

1	1	0	0	0	1	1	1

(A) ← (PSW)

The contents of the program status word (PSW) are moved into the accumulator.

Cycles: 1
Bytes: 1

PSW

Carry	Aux. Carry	Flag 0	Reg. Bank Sel.	1	S2	S1	S0

MSB **LSB**

MOV A, Rr

Move contents of designated register into accumulator.

1	1	1	1	1	r	r	r

(A) ← (Rr); where r = 0 through 7

The contents of the working register designated by bits 'r' are moved into the accumulator.

Cycles: 1
Bytes: 1

MOV A, @ Rr

Move-indirect contents of RAM into accumulator.

1	1	1	1	0	0	0	r

The contents of the internal RAM location, as addressed by bits 0 through 5* of register 'r', are moved to the accumulator.

*bits 0 through 6 for INS8039/INS8049
*bits 0 through 7 for INS8040/INS8050
Cycles: 1
Bytes: 1

MOV A,T

Move contents of timer/counter into accumulator.

0	1	0	0	0	0	1	0

(A) ← (T)

The contents of the timer/counter are moved into the accumulator.

Cycles: 1
Bytes: 1

MOV PSW,A

Move contents of accumulator into program status word.

1	1	0	1	0	1	1	1

(PSW) ← (A)

The contents of the accumulator are moved into the program status word (PSW). All condition bits and status codes are affected by this instruction.

Cycles: 1
Bytes: 1

Flags: Carry, Auxillary Carry, F0, Register Bank Select

PSW

Carry	Aux. Carry	Flag 0	Reg. Bank Sel.	1	S2	S1	S0

MSB LSB

MOV Rr, A

Move accumulator contents into designated register.

1	0	1	0	1	r	r	r

(Rr) ← (A); where r = 0 through 7

The contents of the accumulator are moved into the working register designated by bits 'r'.

Cycles: 1
Bytes: 1

MOV Rr, # data

Move-immediate specified data into designated register.

1	0	1	1	1	r	r	r	byte 1
d7	d6	d5	d4	d3	d2	d1	d0	byte 2

(Rr) ← data; where r = 0 through 7

The data contained in byte 2 is moved into the working register designated by bits 'r' of byte 1.

Cycles: 2
Bytes: 2

MOV @ Rr, A

Move-indirect accumulator contents into RAM.

1	0	1	0	0	0	0	r

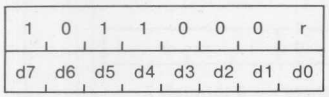
((Rr)) ← (A); where r = 0 or 1

The contents of the accumulator are moved to the RAM location as addressed by bits 0 through 5* of register 'r'. Register 'r' contents are unaffected.

*bits 0 through 6 for INS8039/INS8049
*bits 0 through 7 for INS8040/INS8050
Cycles: 1
Bytes: 1

MOV @ Rr, # data

Move-immediate specified data into RAM.



((R)) ← data; where r = 0 or 1

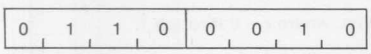
The data contained in byte 2 is moved to the RAM location as addressed by bits 0 through 5* of register 'r'.

*bits 0 through 6 for INS8039/INS8049
*bits 0 through 7 for INS8040/INS8050

Cycles: 2
Bytes: 2

MOV T,A

Move contents of accumulator into timer/counter.



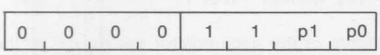
(T) ← (A)

The contents of the accumulator are moved into the timer/counter register.

Cycles: 1
Bytes: 1

MOVD A,Pp

Move contents of designated expansion port (4 through 7) into accumulator.



(A0-3) ← (Pp); where p = 4 through 7
(A4-7) ← 0

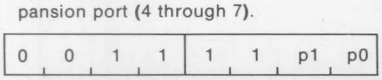
The 4-bit data on the expander port, designated by bits 'p', are moved into accumulator, bits 0 through 3. Bits 4 through 7 in the accumulator are set to zero. Op code bits 'p' designate the following ports:

Port	p1	p0
4	0	0
5	0	1
6	1	0
7	1	1

Cycles: 2
Bytes: 1

MOVD Pp,A

Move contents of accumulator to designated expansion port (4 through 7).



(Pp) ← (A0-3); where p = 4 through 7

The data in the accumulator, bits 0 through 3, are moved to the expander port designated by bits 'p'. Accumulator contents are unaffected.

Op code bits 'p' designate the following ports:

Port	p1	p0
4	0	0
5	0	1
6	1	0
7	1	1

Cycles: 2
Bytes: 1

MOVP A,@A

Move data in current page (ROM) into accumulator.

(PC 0-7) ← (A)
(A) ← ((PC))
(PC 0-7) ← (old PC 0-7) + 1

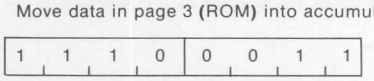
The contents of the program counter, bits 0 through 7, are replaced by the contents of the accumulator. The contents of the internal ROM location, as addressed by the new contents of the program counter, are moved into the accumulator. The program counter is then set back to point to the next sequential instruction. Only bits 0 through 7 of the program counter are affected, limiting memory references to the current page. Being a 1-byte, 2-cycle instruction, if MOVP A is at location 255 of a page, the @ A addresses a location in the following page.

The program counter is restored after the instruction is executed.

Cycles: 2
Bytes: 1

MOVP3 A,@A

Move data in page 3 (ROM) into accumulator.



(PC 0-7) ← (A)
(PC 8-10) ← 011
(A) ← ((PC))
(PC) ← (old PC) + 1

The contents of the program counter, bits 0 through 7, are replaced by the contents of the accumulator. Program counter bits 8 through 10 are replaced by $011_{(2)}$, respectively. The contents of the internal page 3 ROM, as addressed by the new contents of the program counter, are moved into the accumulator. The program is restored after this instruction is executed.

Cycles: 2
Bytes: 1

MOVX A,@R

Move-indirect contents of external RAM into accumulator.

1	0	0	0	0	0	0	r

(A) ← ((Rr)); where r = 0 or 1

The contents of the external RAM location, as addressed by register 'r', are moved into the accumulator. Register 'r' contents are unaffected.

Cycles: 2
Bytes: 1

MOVX @R,A

Move-indirect contents of accumulator into external RAM.

1	0	0	1	0	0	0	r

((Rr)) ← (A); where r = 0 or 1

The contents of the accumulator are moved into the external RAM location, as addressed by register 'r'. Accumulator and register 'r' contents are unaffected.

Cycles: 2
Bytes: 1

NOP

No operation performed.

0	0	0	0	0	0	0	0

No operation is performed; execution continues with the next sequential instruction.

Cycles: 1
Bytes: 1

ORL A,Rr

Logical-OR contents of designated register with accumulator.

0	1	0	0	1	r	r	r

(A) ← (A) OR (Rr); where r = 0 through 7

The contents of the register specified by the 'r' bits are logically ORed with the data in the accumulator.

Cycles: 1
Bytes: 1

ORL A, @R

Logical-OR-indirect contents of RAM with accumulator.

0	1	0	0	0	0	0	r

(A) ← (A) OR ((Rr)); where r = 0 or 1

The contents of the internal RAM location, as addressed by bits 0 through 5* of register 'r', are logically ORed with the data in the accumulator.

*bits 0 through 6 for INS8039/INS8049
*bits 0 through 7 for INS8040/INS8050
Cycles: 1
Bytes: 1

ORL A, #data

Logical-OR specified immediate data with accumulator.

0	1	0	0	0	0	1	1	byte 1
d7	d6	d5	d4	d3	d2	d1	d0	byte 2

(A) ← (A) OR data

The data contained in byte 2 is logically ORed with the data in the accumulator.

Cycles: 2
Bytes: 2

ORL BUS, #data

Logical-OR-immediate specified data with contents of Bus.

1	0	0	0	1	0	0	0	byte 1
d7	d6	d5	d4	d3	d2	d1	d0	byte 2

(BUS) ← (BUS) OR data

The data contained in byte 2 is logically ORed immediately with the data on the Bus port and the results are sent back to that port. Use of this instruction assumes prior execution of an OUTL BUS,A instruction.

Cycles: 2
Bytes: 2

ORL Pp, #data

Logical-OR-immediate specified data with contents of designated Port (1 or 2).

1	0	0	0	1	0	p1	p0	byte 1
d7	d6	d5	d4	d3	d2	d1	d0	byte 2

(Pp) ← (Pp) OR data; where p = 1 or 2

The data contained in byte 2 is logically ORed with the data on the port designated by bits 'p'. Opcode bits 'p' designate the following ports:

Port	p1	p0
1	0	1
2	1	0

Cycles: 2
Bytes: 2

ORLD Pp,A

Logical-OR contents of accumulator with designated expansion port (4-7).

1	0	0	0	1	1	p1	p0

(Pp) ← (Pp) OR (A 0-3); where p = 4 through 7

The data in accumulator bits 0 through 3 are logically ORed with the data on the expander port designated by bits 'p'. Opcode bits 'p' designate the following ports:

Port	p1	p0
4	0	0
5	0	1
6	1	0
7	1	1

Cycles: 2
Bytes: 1

OUTL BUS, A

Output contents of accumulator onto Bus.

0	0	0	0	0	0	1	0

(BUS) ← (A)

The contents of the accumulator are placed, and latched, at the Bus output port. Latched data remains valid until another OUTL BUS instruction is executed, or until another instruction requiring the Bus port (except INS) is executed.

Logical operations using Bus data assume prior execution of the OUTL BUS,A instruction.

Data are destroyed any time the Bus port is used as a bus. The Bus port is in input mode after a reset. After execution of OUTL BUS A, the Bus port will remain an output until the device is either reset or this port is used for bus transfers.

Cycles: 2
Bytes: 1

OUTL Pp,A

Output contents of accumulator to designated port (1 or 2).

0	0	1	1	1	0	p1	p0

(Pp) ← (A); where p = 1 or 2

The contents of the accumulator are placed, and latched, at the output port designated by bits 'p'. Opcode bits 'p' designate the following ports:

Port	p1	p0
1	0	1
2	1	0

Cycles: 2
Bytes: 1

RET

Return from subroutine or interrupt without restoring program status word.

1	0	0	0	0	0	1	1

(SP) ← (SP) -1
(PC) ← ((SP))

The contents of the stack pointer are decremented by one. The contents of the stack, as pointed to by the new contents of the stack pointer, are then placed in the program counter and the program counter is considered restored.

NOTE: *Although the stack pointer is only decremented by one, internally it is decremented by two so the PSW and PC can be pulled off the stack.*

Program status word bits 4 through 7 are not restored by this instruction.

Cycles: 2
Bytes: 1

RETR

Return from subroutine or interrupt restoring program status word.

1	0	0	1	0	0	1	1

(SP) ← (SP) -1
(PC) ← ((SP))
(PSW 4-7) ← ((SP))

The contents of the stack pointer are decremented by one. The contents of the stack, as pointed to by the new contents of the stack pointer, are then placed in the program counter and the program counter is considered restored.

Program status word bits 4 through 7 are restored from the stack to the program status register.

NOTE: *Although the stack pointer is only decremented by one, internally it is decremented by two so the PSW and PC can be pulled off the stack.*

The RETR instruction should be used to return from an interrupt, but should not be used to return from a subroutine within an interrupt. This is because RETR indicates the end of an interrupt routine by re-enabling INT.

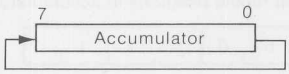
Cycles: 2
Bytes: 1
Flags: Carry, Auxiliary carry, F0, Interrupt Enable Flag (only if Register is servicing an interrupt).

RL A

Rotate accumulator left one bit.

(An + 1) ← (An); where n = 0 through 6
(A0) ← (A7)

The contents of the accumulator are rotated left by one bit position. Bit 7 goes directly to bit 0.



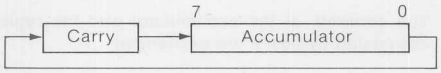
Cycles: 1
Bytes: 1

RLC A

Rotate accumulator left one bit through carry.

(An + 1) ← (An); where n = 0 through 6
(A0) ← (C)
(C) ← (A7)

The contents of the accumulator are rotated left by one bit position. Bit 7 moves to carry and carry moves to bit 0.



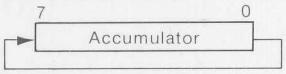
Cycles: 1
Bytes: 1
Flags: Carry

RR A

Rotate accumulator right one bit.

(An) ← (An + 1); where n = 0 through 6
(A7) ← (A0)

The contents of the accumulator are rotated by one bit position.



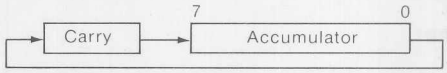
Cycles: 1
Bytes: 1

RRC A

Rotate accumulator right one bit through carry.

(An) ← (An + 1); where n = 0 through 6
(A7) ← (C)
(C) ← (A0)

The contents of the accumulator are rotated right by one bit position. Bit 0 moves to carry and carry moves to bit 7.



Cycles: 1
Bytes: 1
Flags: Carry

SEL MB0

Select bank 0 from ROM (locations 0 through 2047).

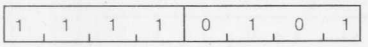
(DBF) ← 0

The memory bank flip-flop (DBF) is set to zero. For succeeding JMP or CALL instructions, program counter bit 11 is set to a logic zero, causing all ROM addresses to fall within locations 0 and 2047.

Cycles: 1
Bytes: 1

SEL MB1

Select bank 1 of ROM (locations 2048 through 4095).



(DBF) ← 1

The memory bank flip-flop (DBF) is set to one. For succeeding JMP or CALL instructions, program counter bit 11 is set to a logic one causing all ROM addresses to fall within locations 2048 and 4095.

Cycles: 1
Bytes: 1

SEL RB0

Select register bank 0 of RAM (locations 0 through 7).

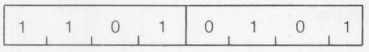
(BS) ← 0

The bank select bit, program status word-bit 4, is set to a logic zero. All references to registers 0 through 7 address RAM locations 0 through 7, respectively. This is the recommended setting for normal program execution.

Cycles: 1
Bytes: 1

SEL RB1

Select register bank 1 of RAM (locations 24 through 31).



(BS) ← 1

The bank select bit, program status word-bit 4 is set to a logic one. All references to registers 0 through 7 address RAM locations 24 through 31, respectively. This is the recommended setting for interrupt service routines. The bank select bit is saved during interrupts and is restored by RETR when the interrupt service routine is completed.

Cycles: 1
Bytes: 1

STOP TCNT

Stop count for timer/counter.

This instruction stops the internal timer/counter, regardless of the mode of operation.

NOTES:
1. If the timer/counter is disabled and one or more falling edges occur at the T1 input, the STRT CNT instruction will cause the counter to increment immediately.
2. There is a mask-programmable resistor option to prevent the preceeding from occurring.

Cycles: 1
Bytes: 1

STRT CNT

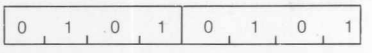
Start count for event counter.

Test input T1 is enabled as the input to the timer/counter and the counter is started. The counter is incremented by one for each high-to-low transition at input T1.

Cycles: 1
Bytes: 1

STRT T

Start timer.



The internal clock is enabled to the timer/counter and the timer is started. The counter is incremented by one for each 32 instruction cycles. The divide-by-32 prescaler is cleared by this instruction.

Cycles: 1
Bytes: 1

SWAP A

Swap 4-bit nibble positions in accumulator.

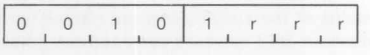
(A 4-7) ↔ (A 0-3)

Accumulator bits 0 through 3 are swapped with accumulator bits 4 through 7.

Cycles: 1
Bytes: 1

XCH A,Rr

Exchange contents of accumulator and designated register.



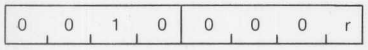
(A) ↔ (Rr); where r = 0 through 7

The contents of the accumulator and the register designated by bits 'r' are exchanged.

Cycles: 1
Bytes: 1

XCH A, @ Rr

Exchange indirect contents of accumulator and RAM location.



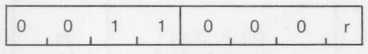
(A) ↔ ((Rr)); where r = 0 or 1

The contents of the internal RAM location, as addressed by bits 0 through 5* of register 'r', are exchanged with the contents of the accumulator. Register 'r' contents are unaffected.

*bits 0 through 6 for INS8039/INS8049
*bits 0 through 7 for INS8050
Cycles: 1
Bytes: 1

XCHD A, @ Rr

Exchange-indirect lower four bits of accumulator and RAM location.



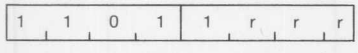
(A 0-3) ↔ (((Rr)) 0-3); where r = 0 or 1

The lower four bits of the internal RAM location, as addressed by bits 0 through 5* of register 'r', are exchanged with the lower four bits of the accumulator. The upper four bits of both RAM and the accumulator are unaffected.

*bits 0 through 6 for INS8039/INS8049
*bits 0 through 7 for INS8040/INS8050
Cycles: 1
Bytes: 1

XRL A,Rr

Logical-XOR contents of designated register with accumulator.



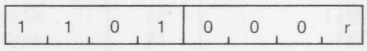
(A) ← (A) XOR (Rr); where r = 0 through 7

The contents of the register specified by bits 'r' are logically EXCLUSIVE-ORed with the data in the accumulator. The register contents are unaffected.

Cycles: 1
Bytes: 1

XRL A, @ Rr

Logical-XOR-indirect contents of RAM with accumulator.



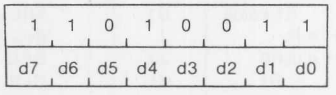
(A) ← (A) XOR ((Rr))

The contents of the internal RAM location, as addressed by bits 0 through 5* of register 'r', are logically EXCLUSIVE-ORed with the data in the accumulator. The internal RAM location contents are unaffected.

*bits 0 through 6 for INS8039/INS8049
*bits 0 through 7 for INS8040/INS8050
Cycles: 1
Bytes: 1

XRL A, #data

Logical-XOR-immediate specified data with accumulator.



(A) ← (A) XOR data

The data contained in byte 2 are logically EXCLUSIVE-ORed with the data in the accumulator.

Cycles: 2
Bytes: 2

4.3 48-SERIES OPERATION CODE

The mnemonic and hexadecimal equivalent for the operation codes (OPCODE) are contained in *Table 4-3*. The table is divided into functional operations.

Table 4-4 through *Table 4-6* provide the hexadecimal opcodes for those instructions that involve registers, parts and pages.

Table 4-3. 48-Series Operation Codes

	MNEMONIC	HEX		MNEMONIC	HEX		MNEMONIC	HEX
	CONTROL			**ACCUMULATOR**			**INPUT/OUTPUT**	
	EN I	05	1	ADD A,R_r	6		IN A,P1	09
	DIS I	15	1	ADD A,@R0	60		OUTL P1,A	39
	SEL RB0	C5		R1	61		ANL P1, #data	99
	SEL RB1	D5	1	ADD A,#data	03		ORL P1, #data	89
	SEL MB0	E5	1 2	ADDC A,R_r	7			
	SEL MB1	F5	1	ADDC A,@R0	70		IN A, P2	0A
	ENT0 CLK	75	1	R1	71		OUT L P2, A	3A
			1	ADDC A,#Data	13		ANL P2 ,#data	9A
	DATA MOVES		2	ANL A,R_r	5		ORL P2, #data	8A
				ANL A,@R0	50			
				R1	51			
2	MOV A,R_r	F		ANL A,#data	53		INS A, BUS	08
	MOV A,@R0	F0	2	ORL A,R_r	4		OUTL BUS, A	02
	R1	F1		ORL A,@R0	40		ANL BUS, #data	98
	MOV A,#data	23		R1	41		ORL BUS, #data	88
2	MOV R,Ar	A						
	MOV @R0,A	A0		ORL A,#data	43	3	MOVD A,Pp	0
	R1,A	A1	2	XRL A,R_r	D	3	MOVD Pp,A	3
2	MOV R_r,#data	B		XRL A,@R0	D0	3	ANLD Pp,A	9
	MOV @R0,#data	B0		R1	D1	3	ORLD Pp,A	8
	R1,#data	B1		XRL A,#data	D3			
2	XCH A,R_r	2		INC A	17			
	XCH A,@R0	20		DEC A	07		**REGISTER**	
	R1	21		CLR A	27			
	XCHD A, @R0	30		CPL A	37	2	INC R_r	1
	R1	31		RL A	E7	2	DEC R_r	C
	MOV A,PSW	C7	1	RLC A	F7		INC @R0	10
1	MOV PSW,A	D7		RR A	77		R1	11
	MOVX A,@R0	80	1	RRC A	67			
	R1	81	1	DA A	57			
	MOVX @R0,A	90		SWAP A	47			
	R1,A	91						
	MOVP3 A,@A	E3						
	MOVP A,@A	A3						

Table 4-3. 48-Series Operation Codes (Cont'd.)

MNEMONIC	HEX		MNEMONIC	HEX		MNEMONIC	HEX
TIMER COUNTER			**BRANCH**			**SUBROUTINE**	
MOV A,T	42	4	JMP addr	4	4	CALL addr	4
MOV T,A	62		JMPP @A	B3		RET	83
STRT T	55	2	DJNZ R_r,addr	E		RETR	93
STRT CNT	45		JC addr	F6			
STOP TCNT	65		JNC addr	E6		**NO OP**	
			JZ addr	C6			
EN TCNTI	25		JNZ addr	96		NOP	00
DIS TCNTI	35		JT0 addr	36			
						FLAGS	
			JNT0 addr	26			
			JT1	56	1	CLR C	97
			JNT1 addr	46	1	CPL C	A7
			JF0 addr	B6		CLR F0	85
			JF1 addr	76		CPL F0	95
			JTI addr	16		CLR F1	A5
			JNI addr	86		CPL F1	B5
			JB0 addr	12			
			JB1 addr	32			
			JB2 addr	52			
			JB3 addr	72			
			JB4 addr	92			
			JB5 addr	B2			
			JB6 addr	D2			
			JB7 addr	F2			

NOTES: 1 = Carry Flag Affected
2 = Refer to Register Accumulator *Table 4-4*
3 = Refer to Input/Output *Table 4-5*
4 = Refer to Branch *Table 4-6*

Table 4-4. Register/Accumulator

R_r	MOV A,R	MOV R,A	XCH A,R	MOV R, #DATA	INC R	DEC R	DJNZ R	ADD A,R	ADDC A,R	ANL A,R	ORL A,R	XRL A,R
R0	F8	A8	28	B8	18	C8	E8	68	78	58	48	D8
R1	F9	A9	29	B9	19	C9	E9	69	79	59	49	D9
R2	FA	AA	2A	BA	1A	CA	EA	6A	7A	5A	4A	DA
R3	FB	AB	2B	BB	1B	CB	EB	6B	7B	5B	4B	DB
R4	FC	AC	2C	BC	1C	CC	EC	6C	7C	5C	4C	DC
R5	FD	AD	2D	BD	1D	CD	ED	6D	7D	5D	4D	DD
R6	FE	AE	2E	BE	1E	CE	EE	6E	7E	5E	4E	DE
R7	FF	AF	2F	BF	1F	CF	EF	6F	7F	5F	4F	DF

Table 4-5. Input/Output

Port	IN	OUT	AND	OR	
BUS	08	02	98	88	48-Series
P1	09	39	99	89	I/O
P2	0A	3A	9A	8A	
P4	0C	3C	9C	8C	INS8243
P5	0D	3D	9D	8D	4-Bit
P6	0E	3E	9E	8E	Ports
P7	0F	3F	9F	8F	

Table 4-6. Branch

Page	JMP	CALL
0	04	14
1	24	34
2	44	54
3	64	74
4	84	94
5	A4	B4
6	C4	D4
7	E4	F4

Page = 256 bytes

Chapter 5

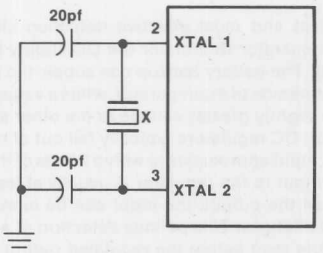
Applications For The 48-Series

5.1 INTRODUCTION

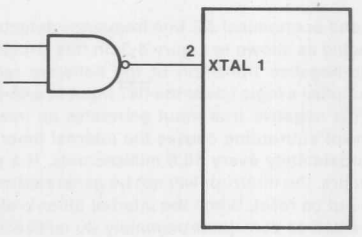
The contents of this chapter are concerned with hardware and software applications. The unique capability of the 48-Series devices require that the examples be integrated with the appropriate driving software. If this were not the case, a hardware example unsupported by any program is of little use. Accordingly, many of the examples that follow are associated with typical software to achieve a particular function.

5.2 TIME BASE INPUTS

Some of the typical 48-Series time-base inputs are illustrated in *Figure 5-1*.



x = 1MHz to 6MHz
or
4MHz to 11MHz
Series Resonant AT Cut
CRYSTAL CONTROLLER



EXTERNALLY CONTROLLED

FIGURE 5-1. 48-Series Time-Base 87-22

5.3 POWER-DOWN OPERATION

Series 48 microcomputers offer the designer a device that is relatively immune to accidental loss of power. These devices provide standby power to selectable areas of RAM when primary power is interrupted. This standby power allows required RAM data to be retained. There are no read or write operations during the standby mode. A power failure can result from low AC line voltage, a complete loss of AC line voltage, or a failure within the system DC power supply. Power failure detection can be performed in the AC circuits or at the output of the system DC power supply.

5.3.1 AC Detection Circuits

Three types of circuits may be used for AC detection: peak detectors, zero-crossing detectors, or line frequency detectors.

5.3.1.1 Peak Detectors

A peak detector (see *Figure 5-2*) consists of a voltage divider, a rectifier circuit, and an AC line isolation capacitor. The voltage divider/rectifier provides a pulsed output that continually retriggers a monostable multivibrator. The output of the multivibrator is used as the power-on/power-fail to the processor.

When the AC line voltage drops to the point where the voltage divider output is below the trigger threshold of the multivibrator, the multivibrator will time out and generate the power-fail signal. (If required, leading and trailing edges of the signal supplied to the multivibrator can be squared using a Schmitt trigger device).

With a peak detector, time is a critical factor. Using full-wave rectification, the maximum power failure detection (retrigger) rate is 8.33 milliseconds. This implies that the filter capacitor, at the output of the rectifier circuit, must be large enough to sustain the power supply output voltage within specification for a minimum of 8.33 milliseconds, if the AC input line drops below specification.

5.3.1.2 Zero-Crossing Detectors

The major draw back of zero-crossing detectors, is that the AC voltage can drop to a level where the DC supply falls out of regulation, yet the detector still detects zero-crossings. This is not acceptable because retrigger signals are applied to the power-fail circuit up to the point where the system stops operation.

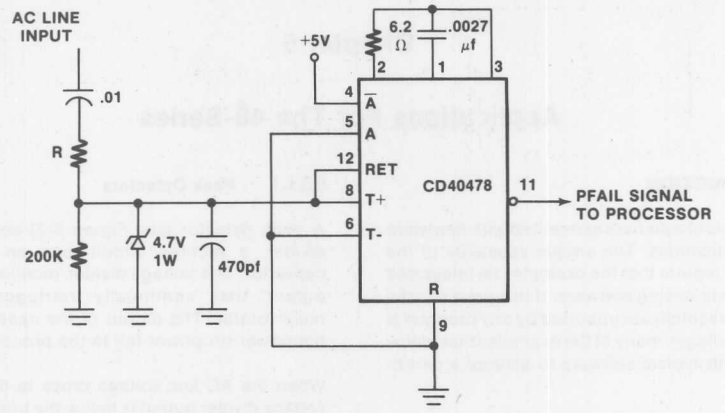


FIGURE 5-2. AC Peak Detection Circuit 87-23

5.3.1.3 Line Frequency Detector

A simple and economical AC line frequency detector can be connected as shown in *Figure 5-3*. In this circuit each positive-to-negative transition of the halfwave rectified AC line supplies a logic low at the $\overline{INT}$ input of a 48-Series device. This negative true input generates an interrupt. The interrupt subroutine causes the internal timer to be reset approximately every 16.6 milliseconds. If a power failure occurs, the interrupt will not be generated and the timer will not be reset. When the internal timer is allowed to count down to zero (approximately 20 milliseconds), a timer interrupt is generated, forcing a jump to a subroutine that performs a power-fail routine.

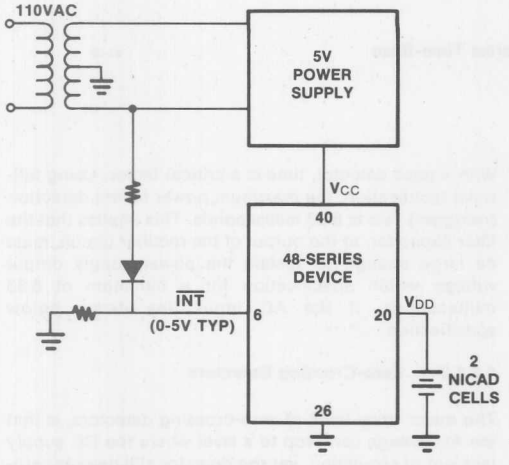


FIGURE 5-3. Line Frequency Detector 87-26

5.3.2 DC Detection

The simplest and most effective detection circuit is a voltage comparator to monitor the DC supply (e.g., see *Figure 5-4*). The battery backup can supply the reference voltage to one side of a comparator, while a voltage divider supplies a slightly greater voltage to the other side of the comparator. DC regulators typically fall out of regulation when their input approaches to within 2 volts of the output. Since the input to the regulator is usually at least 2 volts greater than the output, the input can be monitored for significant changes. This permits detection of a potential power failure long before the regulated output drops.

5.3.3 Power-Fail Interrupt

An imminent power failure would generate an interrupt, causing program operation to branch to a status-save routine. The status-save routine places all data critical to system operation in the standby RAM locations, thereby assuring continuation of the main program when power is restored.

In order for the interrupt to be effective, the enable external interrupt (EN I) instruction must be executed following each system reset. Also, once an interrupt is generated, the interrupt input is disabled until it is enabled by EN I. Saving of status is thus assured while preventing multiple interrupts from a fluctuating power supply.

NOTES:

1. *The user interrupts can be expanded by using the INS8259 Interrupt Controller.*

2. *Expansion can also be achieved by connecting the interrupt request lines to unused input port lines and the ORed sum of the requests into the $\overline{INT}$ input. Source identification can be done by polling the particular port used for interrupt expansion.*

3. *Multiple interrupts can also be generated by using the pullup resistor option for the $\overline{INT}$ input and using open-collector drives to pull the $\overline{INT}$ input down.*

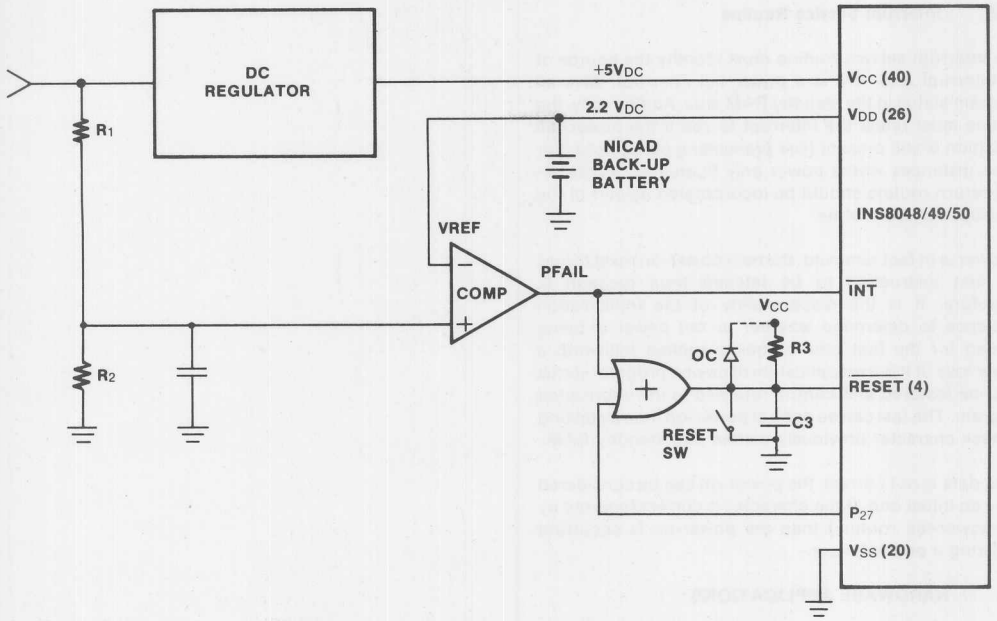


FIGURE 5-4. DC Detection Circuit

87-24

5.3.4 Reset

The reset input should be driven low by the firmware to prevent undesirable system operation during power down. It is important that the state of the power supply be checked after status is saved but before issuing a reset to the microprocessor. The jump-if-not interrupt (JNI) instruction can recheck the interrupt input without creating an interrupt.

If the power supply only fluctuated slightly, this test permits the program to jump around a firmware reset and restore normal program status. If the power fail signal is still active, the reset input should be driven low. An effective method is to drive reset low under firmware control as the final task of the interrupt service routine. If one of the output ports is used to gate the interrupt into the reset input, the microprocessor itself can drive reset low.

Figure 5-5 illustrates how the $\overline{\text{RESET}}$ input may be driven low. When power is initially applied to the system, the Power Fail input to the NAND gate is a logic low (0) and the port pin input to the NAND gate is a logic high (1). The user program sets the port pin input to the gate to a logic low level. In the event of a power failure, the power fail input to both NAND gates becomes a logic high (1) and the $\overline{\text{INT}}$ input to the 48-Series device goes active low (0). The $\overline{\text{INT}}$ input to the device signifies a power failure and the CPU performs a power-fail routine. The power-fail routine saves the machine status and forces the port pin input to the NAND gate to a logic high. With both the power-fail and the port pin inputs at a logic 1 level, the $\overline{\text{RESET}}$ input to the 48-Series device becomes active low preventing the CPU from possibly executing a store instruction to the RAM when power is low.

NOTE: *Since $\overline{\text{RESET}}$ sets all ports high, an active high input should be used to gate $\overline{\text{INT}}$ into $\overline{\text{RESET}}$, otherwise $\overline{\text{RESET}}$ disappears immediately following execution of the instruction that caused it. This also means the bit must be initialized to a zero as part of the power-on sequence.*

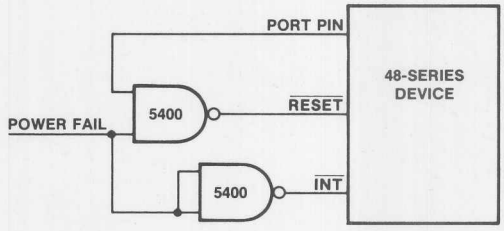


FIGURE 5-5. Driving and Holding $\overline{\text{RESET}}$ Active

87-27

5.3.5 Interrupt Service Routine

The interrupt service routine must identify the source of an interrupt and, if it is a power-fail interrupt, save all program status in the standby RAM area. Additionally, the routine must retest the interrupt to see if the power-fail condition is still present (see preceeding paragraph). For these instances where power only fluctuates, a restore-and-return routine should be incorporated as part of the interrupt service routine.

If power is in fact removed, the next power-on reset forces the first instruction to be fetched from location 0. Therefore, it is the responsibility of the initialization sequence to determine whether or not power is being applied for the first time or being applied following a power loss. If it is a reapplication of power, program status must be restored and control returned to the interrupted program. The test can be done at power-on time by testing a check character previously written into standby RAM.

If the data is not correct, the power-on can be considered to be an initial one. If the character is correct (entered by the power-fail routine) then the power-on is occurring following a power failure.

5.4 HARDWARE APPLICATIONS

Typical 48-Series hardware applications are contained in the INS8048 Series Data Sheet in Appendix A. Additional applications will be included as they are designed.

CHAPTER 6

DEVELOPMENT SUPPORT

6.1 INTRODUCTION

National Semiconductor supports its microprocessors and microcomputers with a full range of publications, technical support, and products.

National's publications contain detailed component or system information; National's technical support consists of full-time training, technical support specialists, and Field Applications Engineers; and National's product support provides an interactive, versatile, and easy-to-use development system.

6.2 PUBLICATIONS

Publications are available covering the various devices manufactured by National Semiconductor. The available literature is grouped in the following categories:

- Literature Index
- Handbooks
- Manuals
- Linear Applications, Vol I and II
- Databooks
- Guides
- Product Selection Guides
- Briefs
- Individual Application Notes
- Individual Data Sheets

See appendix B for more detailed reference material. For list of currently available literature, refer to the Literature Index.

6.3 TRAINING

National Semiconductor operates a microprocessor training center in Santa Clara, Calif. The training center is fully equipped and professionally staffed to provide students with an effective mixture of hardware/software theory and hands-on laboratory experience. Courses covering our microprocessor related products are available at the National Semiconductor Training Center. To obtain information on current courses being offered and schedules, please contact:

Western Training Center
1333 Lawrence Expressway
Santa Clara, CA. 95051
(408) 737-6453

6.4 TECHNICAL SUPPORT PROGRAMS

National Semiconductor has the strongest on-the-scene technical support team - in the U.S. and abroad - of any semiconductor manufacturer. Our large network of independent sales representatives and franchised distributors is backed by our Field Application Engineers (FAE's) and microprocessor Application Engineers. The FAE's are available domestically and internationally to offer on-site technical assistance, and are equipped technically to help analyze your application, translate your needs into a viable hardware/software configuration, and then follow it through to system delivery. The microprocessor application engineers are National's home-base technical support specialists who support the FAE's in the field, and who help you use your microprocessor most effectively; they are always available to answer specific technical questions regarding the use of National's microprocessor and peripheral components.

6.5 STARPLEX DEVELOPMENT SYSTEM

The STARPLEX™ Development System is a general purpose microcomputer and microprocessor development system. New levels of operating simplicity have been designed into the STARPLEX system to significantly reduce the amount of time spent on product development. By getting the user into actual application work sooner and with fewer mistakes, the STARPLEX system allows the user to take full advantage of time spent at the console.

The STARPLEX design combines all the components required for the entire development task in one complete system. The STARPLEX package includes an INS8080-based CPU board, 64K bytes of RAM, 512K bytes of disc storage, a video monitor, keyboard and printer. The standard STARPLEX software package includes a disc operating system, assembler, debugger, editor, linker, loader, FORTRAN, BASIC, on-board ROM diagnostic and utilities. Two options are available: an in-system emulator for real-time debugging of customized hardware and software and a PROM programmer personality module for programming, verifying and copying PROM's.

STARPLEX is illustrated in *Figure 6-1*.

The STARPLEX System reduces the time a user must spend at a terminal by making many complex functions accessible through one easy keystroke. System commands are initiated by clearly marked function keys which invoke prompting menus to guide the user through each task. These function keys eliminate the need to memorize system commands and various command options. As a result, there is no need to refer to lengthy documentation, and errors or delays caused by incorrectly entered commands are eliminated.

Recognizing that a great deal of the user's time is spent on creating and changing source code, the designers of the STARPLEX System have devoted special attention to the text editing facility.

A set of special function keys directs the STARPLEX editor, allowing corrections to be made with single keystrokes. An entire file may be quickly and easily reviewed or altered. The number of mistakes is reduced because the data and changes are immediately displayed.

Backup files are automatically created, protecting the user from accidental loss of data. Because the STARPLEX™ System is easy to use, learning time is considerably shortened. A first time user can be productive within a half hour. Also, as users make more efficient use of the system, machine availability is maximized.

STARPLEX components are packaged into modules which form a unified system when placed together. The modules are durable, with housings constructed of 1/8 inch aluminum and front panels of molded lexan foam.

STARPLEX is designed for easy maintenance. Snap-down doors on the base module make it easy to access the card cages and circuit boards. Interconnecting cables between all modules and boards are routed to the rear of the system and covered by easily removable cable channels. Thus, cables are out of sight and protected from accidental damage. All cables, including the single AC power distribution system, are plug detachable at both ends, making it easy to disconnect modules and reconfigure the system.

Human engineering concepts have directed the design of each STARPLEX module to make the man-machine interface as natural as possible. For example, the video monitor screen has antireflective coating to minimize glare, and light-emitting diodes in certain keys provide operator awareness of their selection. Even cooling fans have been located to minimize noise levels.

More detailed information on STARPLEX, its associated software, and ancillary support cards for STARPLEX, is available in the National Semiconductor Microcomputer Systems Series-80 Databook.

6.6 IN-SYSTEM EMULATOR

National Semiconductor's In-System Emulator, or ISE, goes beyond the single-card approach to emulation and qualifies as a genuine innovation in the development of microprocessor-based systems.

ISE is a complete stand-alone unit that has its own very brief language and can be operated from a simple CRT terminal. The ISE unit contains 32K bytes of user programmable memory and all the necessary logic for breakpoints, tracing and memory mapping. Microprocessor emulation is isolated on a single target card containing all the logic needed to emulate the particular microprocessor. ISE is capable of supporting two of these target cards concurrently to achieve emulation in a multiprocessor environment. ISE can support either two target cards for the same microprocessor or two different microprocessors.

There are three important advantages to a stand-alone emulation system over the emulation card approach:

Performance is the primary advantage. An emulation card must share the host system bus and memory. The card not only shares these resources; it also must compete for them in a priority scheme designed into the host system. This creates an unpredictable environment, making real-time emulation impossible.

In contrast, ISE as a stand-alone system has its own special bus designed for high speed emulation. It also has memory dedicated to the user's program, thus eliminating any conflicts and allowing real-time emulation.

Economy is another advantage of the system approach to emulation. The only difference between one emulator card and another is the microprocessor under emulation. The extensive trace memory, breakpoint logic, memory mapping logic, etc., are the same for all microprocessor emulations. The ISE module contains all the logic common to the emulation process while individual target cards are dedicated to the emulation of particular microprocessors. Each target card supported by ISE shares the total system resources, thus eliminating the unnecessary cost of supplying separate logic and memory on each emulator card.

Convenience is the most obvious advantage. The user needs to master only one software package — either the basic system controller or a more powerful STARPLEX software driver program — both of which suport all features of ISE and a variety of target cards. Specific characteristics of the emulated microprocessor that must be known by the driver program (register complement, word size, status bits, etc.) are recorded in an architecture ROM located on the target card. The driver program simply reads the contents of the architecture ROM when the system is initialized. It then knows which microprocessor it is emulating and the characteristics of that microprocessor.

The ISE software package is totally integrated into the STARPLEX Development System. All of the ease-of-use concepts that set STARPLEX above other development systems are designed into the ISE System.

ISE is called with a single keystroke on the STARPLEX keyboard, as are all other STARPLEX system resources. A fill-in-the-blank menu appears on the CRT and prompts the user to select the microprocessor to be emulated. During the emulation process a portion of the CRT screen is reserved to inform the user of emulation status. This status information includes the type of microprocessor(s), whether they are running, selected, or present, breakpoint condition masks, and whether breakpoints are enabled.

Should the user wish to review the full range of ISE commands available he can call for "HELP", the "HELP" key on the STARPLEX keyboard allows the user to display information describing the ISE software functions.

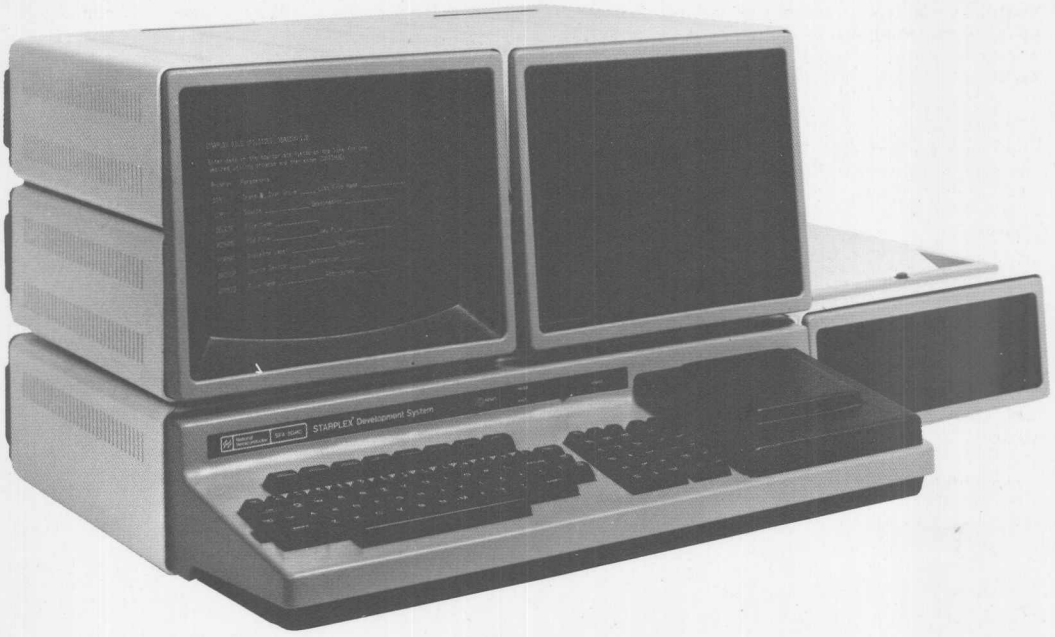
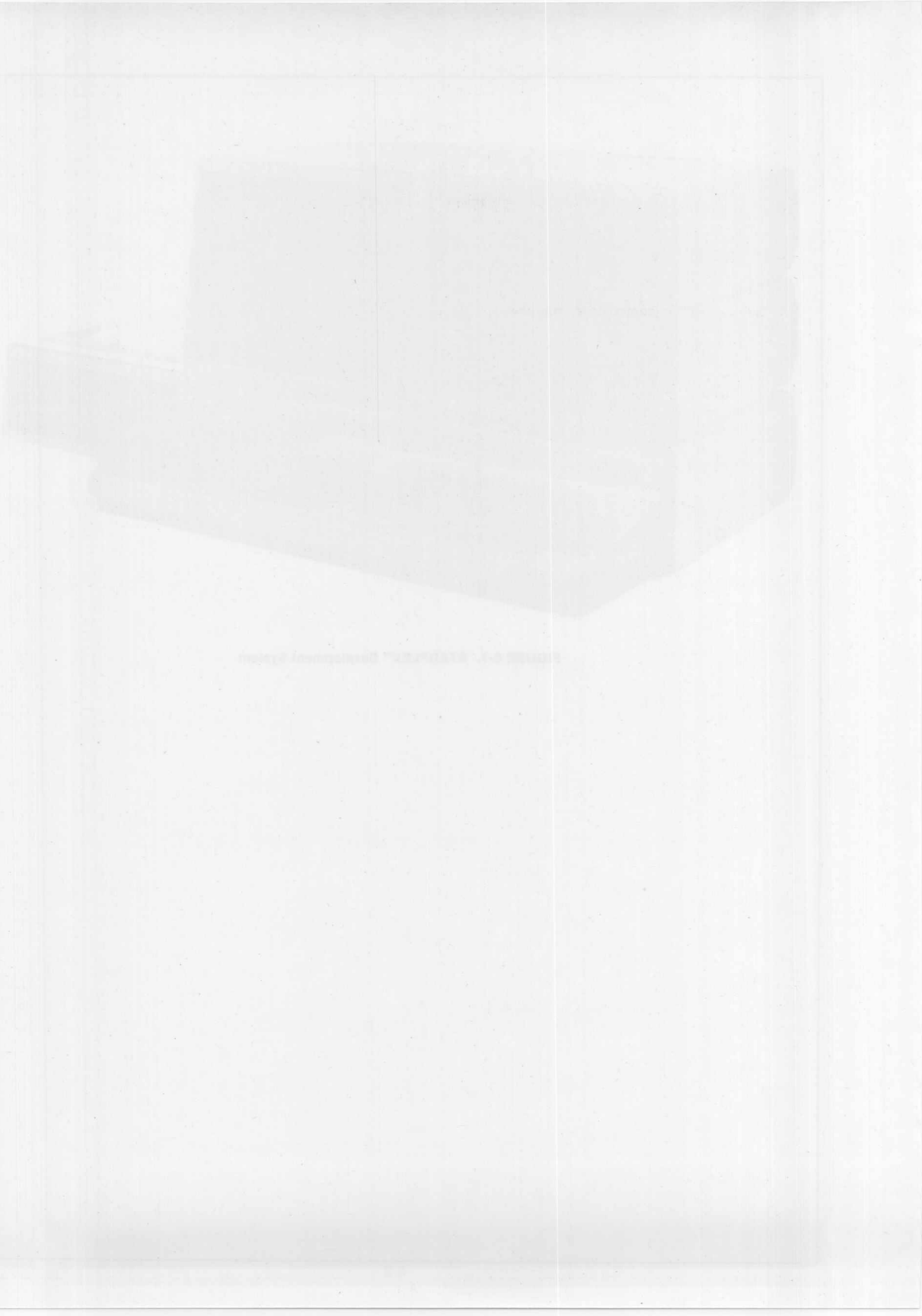


FIGURE 6-1. STARPLEX™ Development System



Appendix A

Data Sheets

Appendix A contains a 48-Series data sheet followed by some compatible device data sheets. The device data sheets are arranged by function.

National Semiconductor

October 1980

INS8048-Series Microcomputer/Microprocessor Family

General Description

The INS8048/49/50-Series microcomputers and the INS8035/39/40-Series microprocessors (hereinafter referred to as the 48-Series) are self contained, 8-bit parallel, 40-pin, dual in-line devices fabricated using National Semiconductor's scaled N-channel, silicon gate MOS process, XMOS. The 48-Series devices contain the system timing, control logic, ROM (where applicable) program memory, RAM data memory and 27 I/O lines necessary to implement dedicated control functions. All 48-Series devices are pin compatible, differing only in the size of on-board ROM (where applicable) and RAM as shown below:

DEVICE	RAM ARRAY	ROM ARRAY
INS8048	64 x 8	1K x 8
INS8049	128 x 8	2K x 8
INS8050	256 x 8	4K x 8
INS8035	64 x 8	N/A
INS8039	128 x 8	N/A
INS8040	256 x 8	N/A

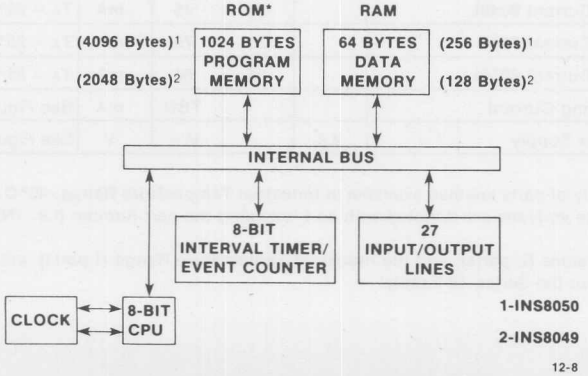
The devices are designed to be efficient controllers. They have extensive bit handling capability as well as facilities for both binary and BCD arithmetic. Efficient use of program memory is derived from an instruction set comprised predominantly of single bytes. The remaining instructions are two bytes in length. Additional external memory may be added up to a maximum of 4K bytes of program memory and 256 bytes of data memory without paging.

Features

- 8-Bit CPU, RAM, ROM, I/O in Single Package
- 2.5 μsec Cycle, 6 MHz Clock; 1.36 μsec Cycle, 11 MHz Clock
- On-Chip Oscillator Circuit and Clock (or External Source)
- 27 I/O Lines
- Expandable Memory and I/O
- 8-Bit Timer/Counter
- Single Level Interrupt
- Interrupt has Schmitt Trigger with Hysteresis*
- Over 90 Instructions (Most Single Byte)
- Binary and BCD Arithmetic
- Single +5V Power Supply
- Low Standby Power Mode*
- Low Voltage Standby (2.2V Min)*
- On-Chip Battery Charging*

*NOTE: *Transparent improvements over industry standard part.*

48-Series Block Diagram



*Not Applicable to INS8035/39/40

Absolute Maximum Ratings

Temperature Under Bias ..... 0°C to +70°C
Storage Temperature..... -65°C to +150°C
All Input or Output Voltages with respect V_{SS} ..... -0.5V to +7.0V
Power Dissipation..... 1.5 Watt

NOTE: *Absolute maximum ratings indicate limits beyond which permanent damage may occur. Continuous operation at these limits is not intended; operation should be limited to those conditions specified under DC Electrical Characteristics.*

DC Electrical Characteristics

T_A = 0° C to +70° C, V_{CC} = +5V ±10%, V_{SS} = 0V, unless otherwise specified.

Symbol	Parameter	Min	Typ	Max	Units	Test Conditions
V_{IL}	Input Low Voltage (All except XTAL1, XTAL2)	-0.5		0.8	V	
V_{IL1}	Input Low Voltage (XTAL1, XTAL2)	-0.5		0.6	V	
V_{IH}	Input High Voltage	2.0		V_{CC}	V	
V_{IH1}	Input High Voltage ($\overline{RESET}$, XTAL1, XTAL2)	3.0		V_{CC}	V	
V_{OL}	Output Low Voltage			0.4	V	I_{OL} = 2.0mA
V_{OH}	Output High Voltage All except ports 1 and 2	3.0		V_{CC}	V	I_{OH} = 100 μA
V_{OH1}	Port TTL	2.4			V	$I_{OH} \geq$ 125 μA
I_{IL}	Input Leakage Current (T1, EA, $\overline{INT}$)			±10	μA	$V_{SS} \leq V_{IN} \leq V_{CC}$
I_{OL}	Output Leakage Current (BUS, T0) (High Impedance State)			-10.0	μA	$V_{CC} \geq V_{IN} \geq V_{SS}$ + 0.45
$I_{DD\ (64)}$	64 words on Standby Current (2)			2.5	mA	8048
$I_{DD\ (128)}$	128 words on Standby Current (2)			4.5	mA	8049
$I_{DD\ (256)}$	256 words on Standby Current (2)			8.5	mA	8050
$I_{DD} + I_{CC}$	Total Supply Current 8048		30	65	mA	T_A = 25°C
$I_{DD} + I_{CC}$	Total Supply Current 8048L		25	40	mA	T_A = 25°C
$I_{DD} + I_{CC}$	Total Supply Current 8049		32	70	mA	T_A = 25°C
$I_{DD} + I_{CC}$	Total Supply Current 8049L		30	45	mA	T_A = 25°C
$I_{DD} + I_{CC}$	Total Supply Current 8050		35	75	mA	T_A = 25°C
$I_{DD} + I_{CC}$	Total Supply Current 8050L		35	50	mA	T_A = 25°C
I_{DDC}	Battery Charging Current			TBD	mA	See *Figure 5*
V_{DD}	Standby Power Supply	2.4		V_{CC}	V	See *Figure 5*

Notes: 1. The Series-48 family of parts are also available in Industrial Temperature Range -40°C to +80°C. Industrial Temperature Range versions are denoted with an I following the part number (i.e., INS8048-6XXX/NI).

2. The low power versions (L parts), and the Industrial Temperature Range (I parts), are currently available throughout the Series-48 Family.

AC Electrical Characteristics - INS80XX-6 (1-6 MHz part)

T_A = 0° C to +70° C, V_{CC} = +5V ±10%, V_{SS} = 0V, unless otherwise specified.

Symbol	Parameter	Min	Typ	Max	Units	Test Conditions
t_{LL}	ALE Pulse Width	400			ns	Note 1
t_{AL}	Address Setup to ALE	150			ns	Note 1
t_{LA}	Address Hold from ALE	80			ns	Note 1
t_{CC}	Control Pulse Width $\overline{PSEN}$, $\overline{RD}$, $\overline{WR}$	700			ns	Note 1
t_{DW}	Data Set-Up Before $\overline{WR}$	500			ns	Note 1
t_{WD}	Data Hold After $\overline{WR}$	120			ns	C_L = 20 pF
t_{CY}	Cycle Time	2.5		15.0	μs	1 to 6 MHz XTAL
t_{DR}	Data Hold	0		200	ns	Note 1
t_{RD}	$\overline{PSEN}$, $\overline{RD}$ to Data In			500	ns	Note 1
t_{AW}	Address Setup to $\overline{WR}$	230			ns	Note 1
t_{AD}	Address Setup to Data In			950	ns	Note 1
t_{AFC}	Address Float to $\overline{RD}$, $\overline{PSEN}$	0			ns	Note 1
t_{CA}	Control Pulse to ALE	10			ns	Note 1

Port 2 Timing

Symbol	Parameter	Min	Typ	Max	Units	Test Conditions
t_{CP}	Port Control Setup before Falling Edge of PROG	110			ns	Note 1
t_{PC}	Port Control Hold after Falling Edge of PROG	140			ns	Note 1
t_{PR}	PROG to Time P2 Input must be Valid			810	ns	Note 1
t_{DP}	Output Data Setup Time	250			ns	Note 1
t_{PD}	Output Data Hold Time	65			ns	Note 1
t_{PF}	Input Data Hold Time	0		150	ns	Note 1
t_{PP}	PROG Pulse Width	1510			ns	Note 1
t_{PL}	Port 2 I/O Data Setup	400			ns	Note 1
t_{LP}	Port 2 I/O Data Hold	150			ns	Note 1

AC Electrical Characteristics - INS80XX-11 (4-11 MHz Part)

T_A = 0° C to +70° C, +5V ±10%, V_{SS} = 0V, unless otherwise specified.

Symbol	Parameter	Min	Typ	Max	Units	Test Conditions
t_{LL}	ALE Pulse Width	150			ns	Note 1
t_{AL}	Address Setup to ALE	70			ns	Note 1
t_{LA}	Address Hold from ALE	50			ns	Note 1
t_{CC}	Control Pulse Width $\overline{PSEN}$, $\overline{RD}$, $\overline{WR}$	300			ns	Note 1
t_{DW}	Data Set-Up Before $\overline{WR}$	250			ns	Note 1
t_{WD}	Data Hold After $\overline{WR}$	40			ns	C_L = 20 pF
t_{CY}	Cycle Time	1.36		3.75	μs	4 to 11 MHz XTAL
t_{DR}	Data Hold	0		100	ns	Note 1
t_{RD}	$\overline{PSEN}$, $\overline{RD}$ to Data In			200	ns	Note 1

AC Electrical Characteristics - INS80XX-11 (Cont'd.)

Symbol	Parameter	Min	Typ	Max	Units	Test Conditions
t_{AW}	Address Set-Up to $\overline{WR}$	200			ns	Note 1
t_{AD}	Address Set-up to Data In			400	ns	Note 1
t_{AFC}	Address Float to $\overline{RD}$, $\overline{PSEN}$	-10			ns	Note 1
t_{CA}	Control Pulse to ALE	10			ns	Note 1

Port 2 Timing

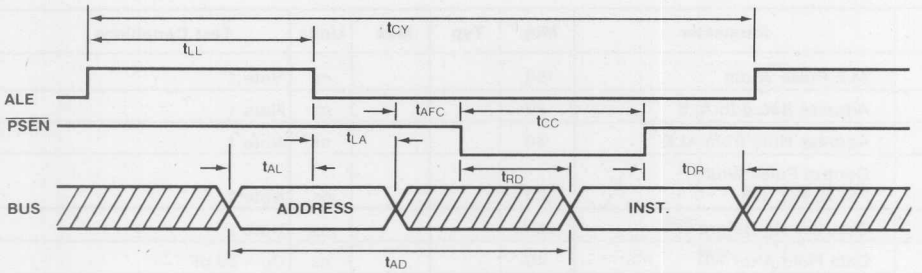
Symbol	Parameter	Min	Typ	Max	Units	Test Conditions
t_{CP}	Port Control Setup before Falling Edge of PROG	100			ns	Note 1
t_{PC}	Port Control Hold after Falling Edge of PROG	60			ns	Note 1
t_{PR}	PROG to Time P2 Input must be Valid			650	ns	Note 1
t_{DP}	Output Data Setup Time	200			ns	Note 1
t_{PD}	Output Data Hold Time	20			ns	Note 1
t_{PF}	Input Data Hold Time	0		150	ns	Note 1
t_{PP}	PROG Pulse Width	700			ns	Note 1
t_{PL}	Port 2 I/O Data Setup	150			ns	Note 1
t_{LP}	Port 2 I/O Data Hold	20			ns	Note 1

Note 1. Control outputs C_L = 80 pF; Bus outputs C_L = 150 pF

Capacitance $T_A = 25°$ C, $V_{CC} = V_{SS} = 0V$

Symbol	Parameter	Min	Typ	Max	Units	Test Conditions
C_{IN}	Input Capacitance		6	10	pF	f_c = 1 MHz
C_{OUT}	OUTPUT AND $\overline{RESET}$ Capacitance		10	20	pF	Unmeasured pins returned to V_{SS}

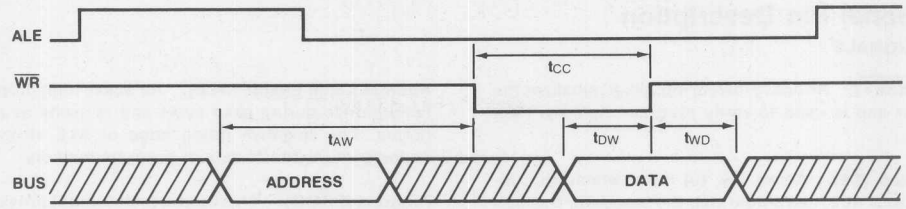
Timing Waveforms



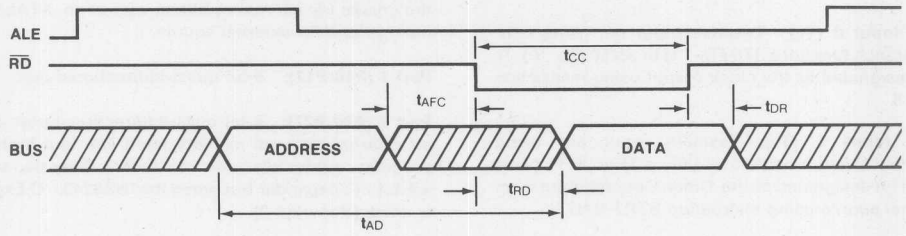
12-14

Instruction Fetch from External Program Memory

NOTE: Diagonal lines indicate interval of high impedance.

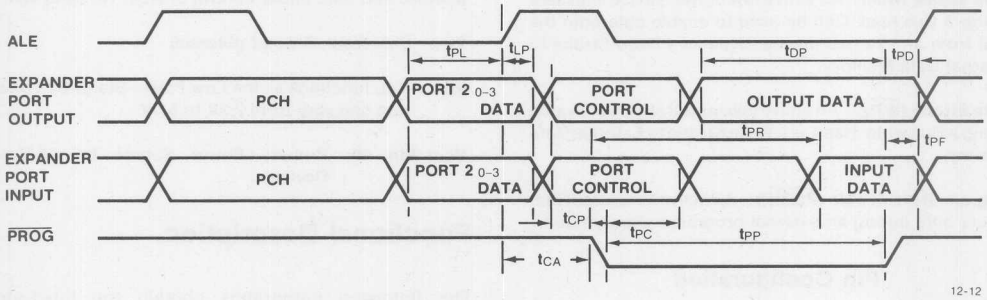


Write to External Data Memory 12-15



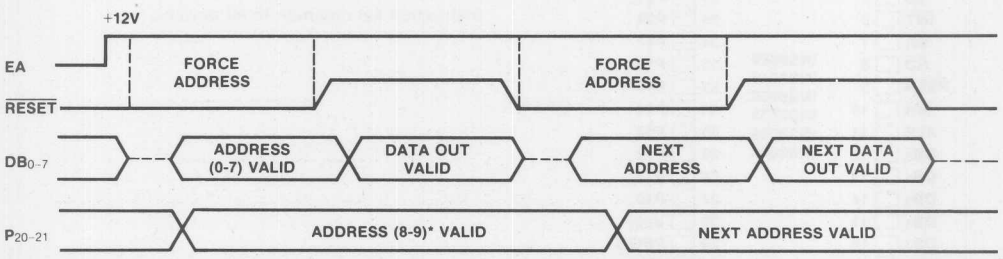
Read from External Data Memory

NOTE: Diagonal lines indicate interval of high impedance. 12-11



12-12

Port 2 Timing



*8049 = 8-10
8050 = 8-11

Verify Mode Timing 12-13

Functional Pin Description

INPUT SIGNALS

Reset ($\overline{\text{RESET}}$): An active low (0) input that initializes the processor and is used to verify program memory. (See note #1.)

Single Step ($\overline{\text{SS}}$): Active low (0) input which, in conjunction with ALE, can single step the processor through each instruction.

External Access (EA): An active high (1) input that forces all program memory fetches to reference external program memory.

Testable Input 0 (T_0): Testable input pin using conditional branch functions JT0 (T_0 1) or JNT0 (T_0 0). T_0 can be designated as the clock output using instruction ENT0 CLK.

Testable Input 1 (T_1): Testable input pin using conditional branch functions JT1 (T_1 = 1) or JNT1 (T_1 = 0). T_1 can be designated as the Timer/Counter input from an external source using instruction STRT CNT.

Interrupt ($\overline{\text{INT}}$): An active low input that initiates an interrupt when interrupt is enabled. Interrupt is disabled after a reset. Also can be tested with instruction JNI (INT 0). (See Note 2).

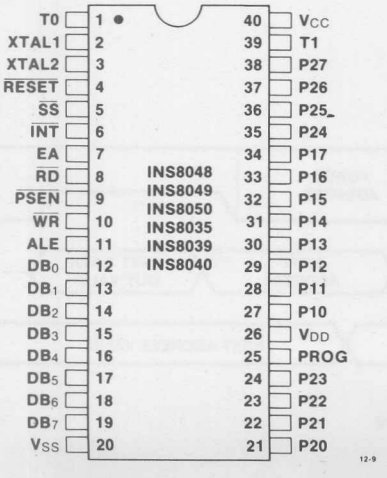
OUTPUT SIGNALS

Read Strobe ($\overline{\text{RD}}$): An active low output strobe activated during a Bus read. Can be used to enable data onto the BUS from an external device. Used as a Read Strobe to External Data Memory.

Write Strobe ($\overline{\text{WR}}$): An active low output strobe activated during a Bus write. Used as a Write Strobe to External Data Memory.

Program Store Enable ($\overline{\text{PSEN}}$): An active low output that occurs only during an external program memory fetch.

Pin Configuration



Address Latch Enable (ALE): An active high output that occurs once during each cycle and is useful as a clock output. The negative going edge of ALE strobes the address into external data or program memory.

Program (PROG): This output (active high) provides the output strobe for INS8243 I/O Expander.

INPUT/OUTPUT SIGNALS

Crystal Input (XTAL1, XTAL2): These two pins con-nect the crystal for internal oscillator operation. XTAL1 is the timing input for external source.

Port 1 (P10-P17): 8-bit quasi-bidirectional port.

Port 2 (P20-P27): 8-bit quasi-bidirectional port. During an external program memory fetch, the four high-order program counter bits occur at P20-P23. They also serve as a 4-bit I/O expander bus when the INS8243 I/O Expander is used. (See note 3).

BUS (DB_0-DB_7): True bidirectional port, either statically latched or synchronous. Can be written to using $\overline{\text{WR}}$ Strobe, or Read from using $\overline{\text{RD}}$ Strobe. During an external program memory fetch, the 8 lower order program counter bits are preset at this port. The addressed instruction appears on this bus when $\overline{\text{PSEN}}$ is low. During an external RAM data store instruction. This port presents address and data under control of ALE, $\overline{\text{RD}}$, and $\overline{\text{WR}}$.

V_{SS}: Processor Ground potential.

V_{DD}: V_{DD} functions as the Low Power Stand-by Voltage and can vary from 2.2V to 5.5V.

V_{CC}: Pin 40: Primary Power Source for 48-Series Devices.

Functional Description

The following paragraphs contain the functional description of the major elements of the 48-Series microcomputer/microprocessor. *Figure 1* is a block diagram of the 48-Series devices. The data paths are illustrated in simplified form to show how the various logic elements communicate with each other to implement the instruction set common to all devices.

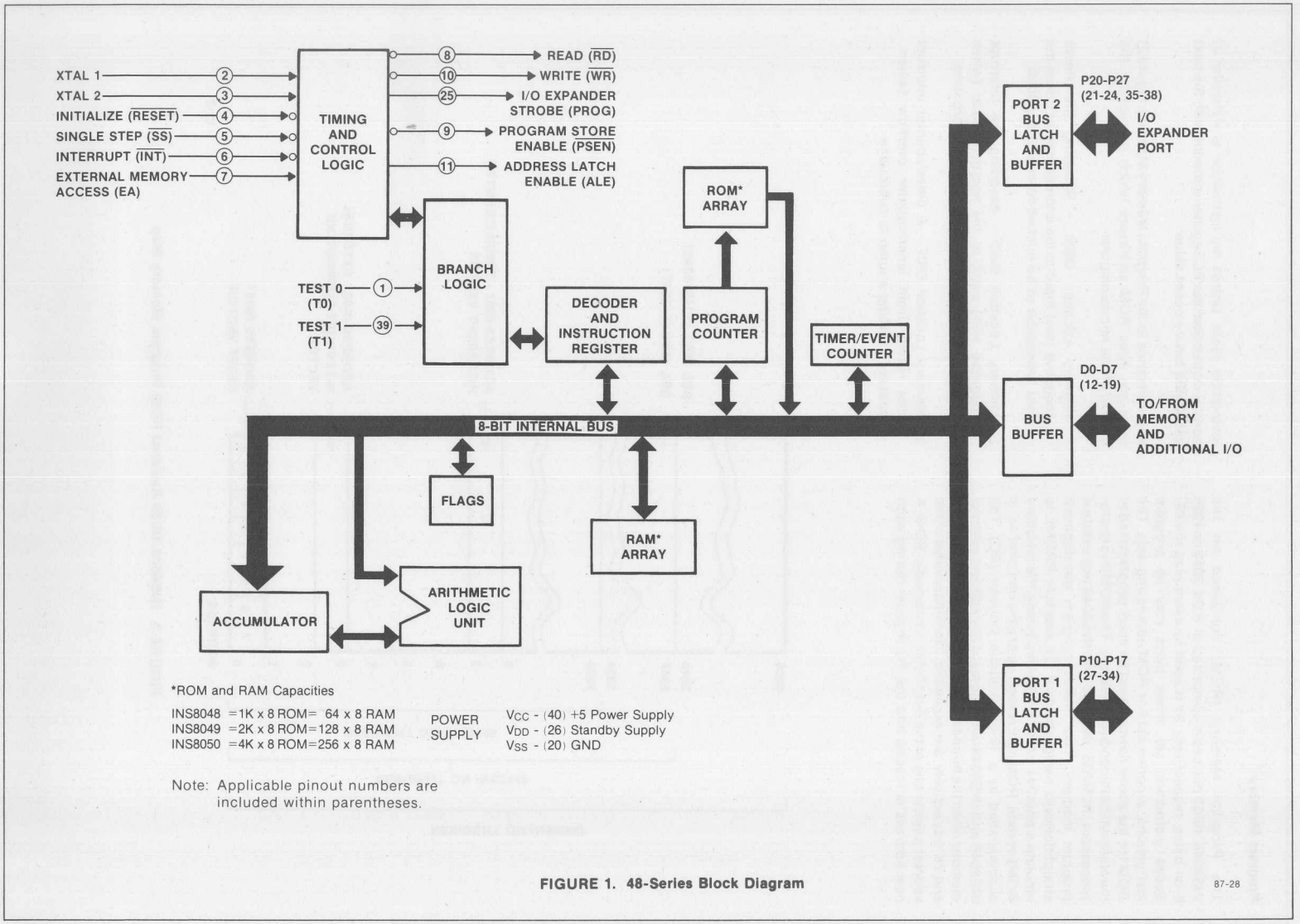


FIGURE 1. 48-Series Block Diagram

87-28

Program Memory

The Program Memory (ROM) contained on the INS8048/49/50 devices is comprised of 1024, 2048 or 4096 8-bit bytes, respectively. As is seen by examining the 48-Series instruction set, these bytes may be program instructions, program data or ROM addressing data. The ROM for the above devices must be mask programmed at the National Semiconductor factory. The ROMless microprocessors, INS8035, INS8039 and INS8040 use external program memory. This makes program development straightforward using standard UV erasable PROMs to emulate a possible future single chip (using the on-board ROM) system. ROM addressing, up to a maximum of 4K, is accomplished by a 12-bit Program Counter (PC). The INS8048 and INS8049 will automatically address external memory when the boundary of their internal memories, 1K and 2K respectively, are exceeded. The binary value of the address selects one of the 8-bit bytes contained in ROM. A new address is loaded into the PC register during each instruction cycle. Unless the instruction is a transfer of control instruction, the PC register is loaded with the next sequential binary count value.

With reference to the Program Memory Map (see *Figure 2*) there are three ROM addresses which provide for the control of the microcomputer.

1. Memory Location 0000 - Resetting the Reset (negative true) input to the microcomputer forces the first instruction to be fetched from address 0000.

2. Memory Location 0003 - Asserting the Interrupt (negative true) input to the microcomputer (when interrupt is enabled) forces a jump to subroutine.

3. Memory Location 0007 - A timer/counter interrupt that results from timer/counter overflow (when enabled) forcing a jump to subroutine.

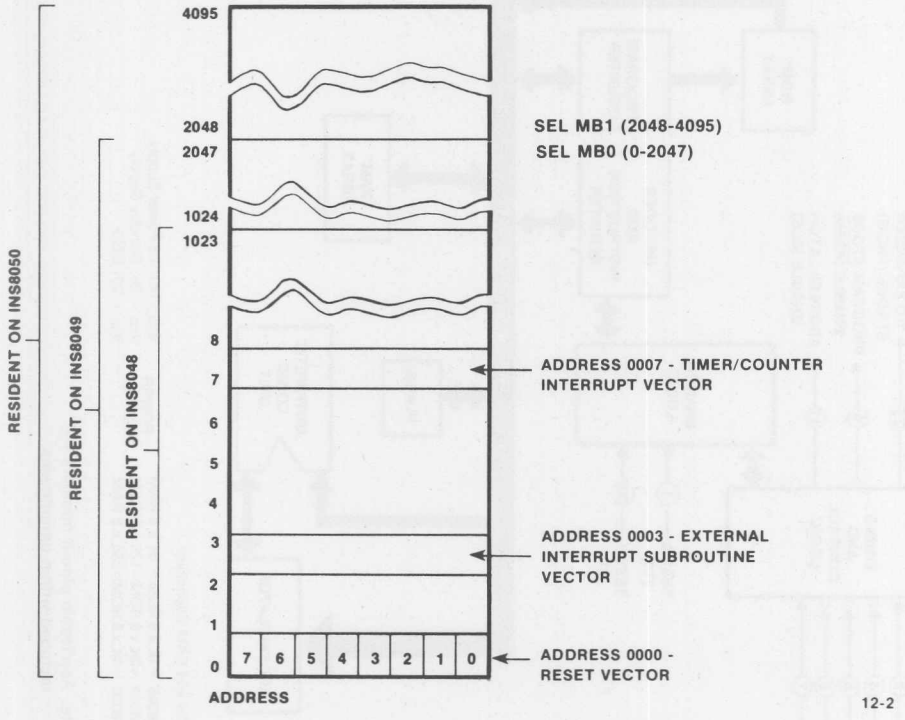


FIGURE 2. INS8048/49/50 Resident ROM Program Memory Map

Data Memory (RAM)

The resident RAM data memory is arranged as 64 (INS8035/8048), 128 (INS8039/8049) or 256 (INS8040/8050) bytes. RAM addressing is implemented indirectly via either of two 8-bit RAM pointer registers R0 and R1. These pointer registers are essentially the first two locations in the RAM (see *Figure 3*), addresses 000 and 001. RAM addressing may also be performed directly by 11 direct register instructions. The pointer register area of the RAM array is made up of eight working registers that occupy either the first bank (0), locations (0 to 7), or the second bank (1), locations 24-31. The second bank of working registers is selected by using the Register Bank Switch instruction (SEL RB). If this bank is not used for working registers, it can be used as user RAM.

There is an 8-level stack after Bank 0 that occupies address locations 8 to 23. These RAM locations are addressed indirectly through R0, R1 or the 3-bit Stack Pointer (SP). The stack pointer keeps track of the return address and pushes each return address down into the stack. There are 8 levels of subroutine nesting possible in the stack because each address occupies 10 bits or more using two bytes in RAM. When the level of subroutine nesting is less than 8, the stacks not used may be utilized as user RAM locations.

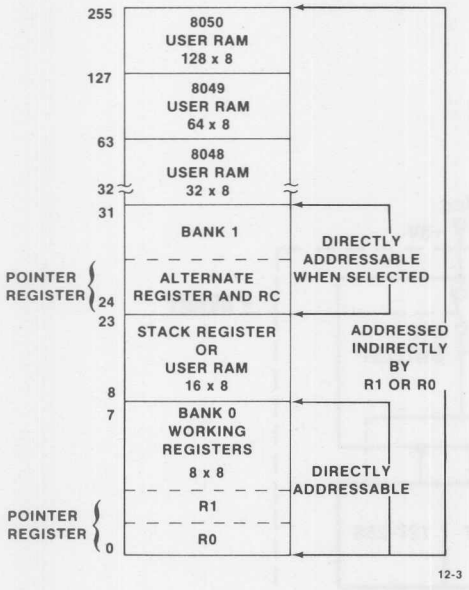


FIGURE 3. 48-Series Resident RAM Data Memory Map

Input/Output

The 48-Series devices have 27 lines of input/output organized as three, 8-bit ports plus three test inputs. The three ports may be used as inputs, outputs or bidirectional ports. Ports 1 and 2 differ from port 3 (Bus Port) in that they are quasi-bidirectional ports. Ports 1 and 2 can be used as input and output while being statically latched. If more I/O lines are required, Port 2 can also serve as a 4-bit I/O bus expander when used in con-junction with the INS8243 I/O Expander.

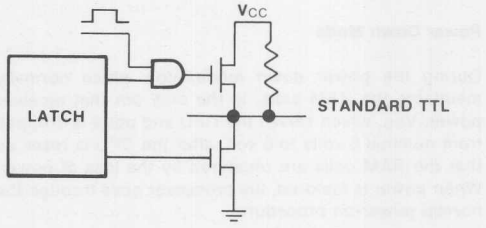


FIGURE 4. Input/Output Options

The bus port is a true bidirectional port and is either statically latched or synchronous. It can be written to using $\overline{WR}$ strobe or read from using $\overline{RD}$ strobe. During an external program memory fetch, the 8 lower order program counter bits are preset at this port. The addressed instruction appears on this bus when $\overline{PSEN}$ is low. During an external RAM data store instruction, this port presents address and data under control of ALE, $\overline{RD}$, and $\overline{WR}$.

Transparent Improvements

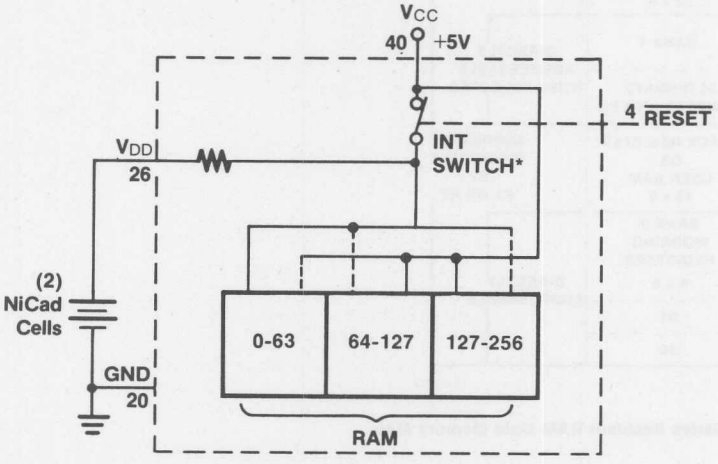
National has made some additional improvements to the standard industry parts. These include a battery charging circuit, and interrupt pin with hysteresis. Also, these improvements are transparent to the user. See *Figure 5.*

Power Down Mode

During the power down mode, V_{DD} which normally maintains the RAM cells, is the only pin that receives power. V_{CC}, which serves the CPU and ports is dropped from nominal 5 volts to 0 volt, after the CPU is reset, so that the RAM cells are unaltered by the loss of power. When power is restored, the processor goes through the normal power-on procedure.

Battery Charging Circuit

All 48-Series devices contain a circuit to provide external battery charging capabilities. Power for all on-board circuits are provided by V_{CC} (pin 40). As shown in *Figure 5* under normal operating conditions the RESET input is a logic high holding the internal switch in the closed position. V_{CC} is supplied to the program selectable portion of the RAM array through the closed contact of the internal switch. The normally closed contacts of the switch also provide charging power to the external NiCad cells. In the event of power failure, the RESET pin must be pulled low before V_{CC} drops below 4.5 volts in order to guarantee the RAM will not loose data. When the RESET pin becomes a logic low (0V) the internal switch is forced to the open condition. DC power to sustain the desired RAM data is provided by the two NiCad batteries (approximately 2.2 volts). Normally, approximately 5 volts are required to provide RAM data protection in the event of a power failure. National's innovative advances in NMOS technology provide the user with a RAM that requires 50% less voltage and 10% of the power to protect data during power failure. The on-chip charging circuit and lower RAM power requirements provide the user with a twofold saving; no external circuits required for the battery charging and only 2 NiCad cells as opposed to the normal requirement of 4 to 5 NiCad cells.



12-5

FIGURE 5. INS8049 Battery Charging Circuit

Instruction Set

Table 1 details the 96 instructions common to both the microcomputers and the microprocessors. The table provides the mnemonic, function and description, instruction code, number of cycles and, where applicable, flag settings.

Table 1 Instruction Set

MNEMONIC	FUNCTION	DESCRIPTION	CYCLES	BYTES	FLAGS C	AC	F0	F1
		CONTROL						
EN I		Enable the External Interrupt input.	1	1				
DIS I		Disable the External Interrupt input.	1	1				
ENT0 CLK		Enable T0 as the Clock Output.	1	1				
SEL MB0	(DBF) ← 0	Select Bank 0 (locations 0 - 2047) of Program Memory.	1	1				
SEL MB1	(DBF) ← 1	Select Bank 1 (locations 2048 - 4095) of Program Memory.	1	1				
SEL RB0	(BS) ← 0	Select Bank 0 (locations 0 - 7) of Data Memory.	1	1				
SEL RB1	(BS) ← 1	Select Bank 1 (locations 24 - 31) of Data Memory.	1	1				
		DATA MOVES						
MOV A, #data	(A) ← data	Move Immediate the specified data into the Accumulator.	2	2				
MOV A, Rr	(A) ← (Rr); r = 0 - 7	Move the contents of the designated registers into the Accumulator.	1	1				
MOV A, @ Rr	(A) ← ((Rr)); r = 0 - 1	Move Indirect the contents of data memory location into the Accumulator.	1	1				
MOV A, PSW	(A) ← (PSW)	Move contents of the Program Status Word into the Accumulator.	1	1				
MOV Rr, #data	(Rr) ← data; r = 0 - 7	Move Immediate the specified data into the designated register.	2	2				
MOV Rr, A	(Rr) ← (A); r = 0 - 7	Move Accumulator contents into the designated register.	1	1				
MOV @ Rr, A	((Rr)) ← (A); r = 0 - 1	Move Indirect Accumulator contents into data memory location.	1	1				
MOV @ Rr, #data	((Rr)) ← data; r = 0 - 1	Move Immediate the specified data into data memory.	2	2				
MOV PSW, A	(PSW) ← (A)	Move contents of Accumulator into the Program Status Word.	1	1	•	•	•	
MOVP A, @ A	(PC 0 - 7) ← (A) (A) ← ((PC))	Move the content of program memory location in the current page addressed by the content of accumulator into the accumulator.	2	1				
MOVP3 A, @ A	(PC 0 - 7) ← (A) (PC 8 - 10) ← 011 (A) ← ((PC))	Move the content of program memory location in page 3 address by the content of accumulator into the accumulator.	2	1				
MOVX A, @ R	(A) ← ((Rr)); r = 0 - 1	Move Indirect the contents of external data memory into the Accumulator.	2	1				

Table 1. Instruction Set (Cont'd.)

MNEMONIC	FUNCTION	DESCRIPTION	CYCLES	BYTES	FLAGS C	AC	F0	F1
		DATA MOVES (Cont'd.)						
MOVX @ R, A	((Rr)) ← (A); r = 0 - 1	Move Indirect the contents of the Accumulator into external data memory.	2	1				
XCH A, Rr	(A) ↔ (Rr); r = 0 - 7	Exchange the Accumulator and designated register's contents.	1	1				
XCH A, @ Rr	(A) ↔ ((Rr)); r = 0 - 1	Exchange Indirect contents of Accumulator and location in data memory.	1	1				
XCHD A, @ Rr	(A0 - A3) ↔ (((Rr)) 0 - 3); r = 0 - 1	Exchange Indirect 4-bit contents of Accumulator and data memory.	1	1				
		TIMER COUNTER						
EN TCNTI		Enable Internal Interrupt Flag for Timer/Counter output.	1	1				
DIS TCNTI		Disable Internal Interrupt Flag for Timer/Counter output.	1	1				
MOV A, T	(A) ← (T)	Move contents of Timer/Counter into Accumulator.	1	1				
MOV T, A	(T) ← (A)	Move contents of Accumulator into Timer/Counter.	1	1				
STOP TCNT		Stop Count for Event Counter.	1	1				
STRT CNT		Start Count for Event Counter.	1	1				
STRT T		Start Count for Timer.	1	1				
		ACCUMULATOR						
ADD A, #data	(A) ← (A) + data	Add Immediate the specified Data to the Accumulator.	2	2	•	•		
ADD A, Rr	(A) ← (A) + (Rr) for r = 0 - 7	Add contents of designated register to the Accumulator.	1	1	•	•		
ADD A, @ Rr	(A) ← (A) + ((Rr)) for r = 0 - 1	Add Indirect the contents the data memory location to the Accumulator.	1	1	•	•		
ADDC A, #data	(A) ← (A) (C) + data	Add Immediate with carry the specified data to the Accumulator.	2	2	•	•		
ADDC A, Rr	(A) ← (A) + (C) + (Rr) for r = 0 - 7	Add with carry the contents of the designated register to the Accumulator.	1	1	•	•		
ADDC A, @ Rr	(A) ← (A) + (C) + ((Rr)) for r = 0 - 1	Add Indirect with carry the contents of data memory location to the Accumulator.	1	1	•	•		
ANL A, #data	(A) ← (A) AND data	Logical AND specified Immediate Data with Accumulator.	2	2				
ANL A, Rr	(A) ← (A) AND (Rr) for r = 0 - 7	Logical AND contents of designated register with Accumulator.	1	1				
ANL A, @ Rr	(A) ← (A) AND ((Rr)) for r = 0 - 1	Logical AND Indirect the contents of data memory with Accumulator.	1	1				
CPL A	(A) ← NOT (A)	Complement the contents of the Accumulator.	1	1				
CLR A	(A) ← 0	CLEAR the contents of the Accumulator.	1	1				
DA A		DECIMAL ADJUST the contents of the Accumulator.	1	1	•			
DEC A	(A) ← (A) - 1	DECREMENT by 1 the accumulator's contents.	1	1				
INC A	(A) ← (A) + 1	Increment by 1 the accumulator's contents	1	1				
ORL A #data	(A) ← (A) OR data	Logical OR specified immediate data with Accumulator.	2	2				
ORL A, Rr	(A) ← (A) OR (Rr) for r = 0 - 7	Logical OR contents of designated register with Accumulator.	1	1				
ORL A, @ Rr	(A) ← (A) OR ((Rr)) for r = 0 - 1	Logical OR Indirect the contents of data memory location with Accumulator.	1	1				

Table 1. Instruction Set (Cont'd.)

MNEMONIC	FUNCTION	DESCRIPTION	CYCLES	BYTES	FLAGS C	AC	F0	F1
		ACCUMULATOR (Cont'd.)						
RLA	(An + 1) ← (An) for n = 0 - 6 (A0) ← (A7)	Rotate Accumulator left by 1-bit without carry.	1	1				
RLC A	(An+1) ← (An); n = 0-6 (A0) ← (C) (C) ← (A7)	Rotate Accumulator left by 1-bit through carry.	1	1	•			
RR A	(An) ← (An+1); n = 0-6 (A7) ← (A0)	Rotate Accumulator right by 1-bit without without carry.	1	1				
RRC A	(An) ← (An+1); n + 0-6 (A7) ← (C) (C) ← (A0)	Rotate Accumulator right by 1-bit through carry.	1	1	•			
SWAP A	(A4-A7) ↔ (A0 - A3)	Swap the 2, 4-bit nibbles in the Accumulator.	1	1				
XRL A, #data	(A) ← (A) XOR data	Logical XOR immediate specified data with Accumulator.	2	2				
XRL A, Rr	(A) ← (A) XOR (Rr) for r = 0 - 7	Logical XOR contents of designated register with Accumulator.	1	1				
XRL A, @ Rr	(A) ← (A) XOR ((Rr)) for r = 0 - 1	Logical XOR Indirect the contents of data memory location with Accumulator.	1	1				
		BRANCH						
DJNZ Rr, addr	(Rr) ← (Rr) -1; r = 0-7 if (Rr) ≠ 0; (PC 0-7) ← addr	Decrement the specified register and test contents.	2	2				
JBb addr	(PC 0-7) ← addr if Bb = 1 (PC) ← (PC) + 2 if Bb = 0	Jump to specified address if Accumulator bit is set.	2	2				
JC addr	(PC 0-7) ← addr if C = 1 (PC) ← (PC) +2 if C = 0	Jump to specified address if carry flag is set.	2	2				
JF0 addr	(PC 0-7) ← addr if F0 = 1 (PC) ← (PC) + 2 if F0 = 0	Jump to specified address if Flag F0 is set.	2	2				
JFI addr	(PC 0-7) ← addr if F1 = 1 (PC) ← (PC) + 2 if F1 = 0	Jump to specified address if Flag F1 is set.	2	2				
JMP addr	(PC 8-10) ← addr 8-10 (PC 0-7) ← addr 0-7 (PC 11) ← DBF	Direct Jump to specified address within the 2K address block.	2	2				
JMPP @ A	(PC 0-7) ← ((A))	Jump indirect to specified address pointed to by the accumulator in current page.	2	1				
JNC addr	(PC 0-7) ← addr if C = 0 (PC) ← (PC) + 2 if C = 1	Jump to specified address if carry flag is low.	2	2				
JNI addr	(PC 0-7) ← addr if I = 0 (PC) ← (PC) + 2 if I = 1	Jump to specified address if interrupt is low.	2	2				
JNT0 addr	(PC 0-7) ← addr if T0 = 0 (PC) ← (PC) + 2 if T0 = 1	Jump to specified address if Test 0 is low.	2	2				
JNT1 addr	(PC 0-7) ← addr if T1 = 0 (PC) ← (PC) + 2 if T1 = 1	Jump to specified address if Test 1 is low.	2	2				
JNZ addr	(PC 0-7) ← addr if A ≠ 0 (PC) ← (PC) + 2 if A = 0	Jump to specified address if accumulator is non-zero.	2	2				
JTF addr	(PC 0-7) ← addr if TF = 1 (PC) ← (PC) + 2 if TF = 0	jump to specified address if Timer Flag is set to 1.	2	2				
JT0 addr	(PC 0-7) ← addr if T0 = 1 (PC) ← (PC) + 2 if T0 = 0	Jump to specified address if Test 0 is a 1.	2	2				
JT1 addr	(PC 0-7) ← addr if T1 = 1 (PC) ← (PC) + 2 if T1 = 0	Jump to specified address if Test 1 is a 1.	2	2				
JZ addr	(PC 0-7) ← addr if A 0 (PC) ← (PC) + 2 if A 1	Jump to specified address if Accumulator is 0.	2	2				

Table 1. Instruction Set (Cont'd.)

MNEMONIC	FUNCTION	DESCRIPTION	CYCLES	BYTES	FLAGS C	AC	F0	F1
		INPUT/OUTPUT						
ANL BUS, data	(BUS) ← (BUS) AND data	Logical AND Immediate specified data with contents of BUS.	2	2				
ANL Pp, data	(Pp) ← (Pp) AND data; p 1 - 2	Logical AND immediate specified data with designated port (1 or 2).	2	2				
ANLD Pp, A	(Pp) ←(Pp) AND (A0 - A3); p 4-7	Logical AND contents of Accumulator with designated port (4 - 7).	2	1				
IN A, Pp	(A) ← (Pp); p 1-2	Input data from designated port (1 - 2) into Accumulator.	2	1				
INS A, BUS	(A) ← (BUS)	Input strobed BUS data into Accumulator	2	1				
MOVD A, Pp	(A0-A3) ← (Pp); p 4-7 (A4-A7) ← 0	Move contents of designated port (4 - 7) into Accumulator.	2	1				
MOVD Pp, A	(Pp) ← (A0 - A3); p 4 - 7	Move contents of Accumulator to designated port (4 - 7).	2	1				
ORL BUS, #data	(BUS) ← (BUS) OR data	Logical OR Immediate specified data with contents of BUS.	2	2				
ORLD Pp, A	(Pp) ← (Pp) OR (A0 - A3); p 4-7.	Logical OR contents of Accumulator with designated port (4 - 7).	2	1				
ORL Pp, #data	(Pp) ← (Pp) OR data; p 1 - 2.	Logical OR Immediate specified data with designated port (1 - 2).	2	2				
OUTL BUS, A	(BUS) ← (A)	Output contents of Accumulator onto BUS.	2	1				
OUTL Pp, A	(Pp) ← (A); p 1 - 2	Output contents of Accumulator to designated port (1 - 2).	1	1				
		REGISTERS						
DEC Rr	(Rr) ← (Rr) -1; r 0-7	Decrement by 1 contents of designated register.	1	1				
INC Rr	(Rr) ← (Rr) + 1; r 0-7	Increment by 1 contents of designated register.	1	1				
INC @ Rr	((Rr)) ← ((Rr)) + 1; r 0-1.	Increment Indirect by 1 the contents of data memory location.	1	1				
		SUBROUTINE						
CALL addr	((SP)) ← (PC) ((SP)) ←(PSW 4-7) (SP) ← (SP) + 1 (PC8-10) ← addr 8-10 (PC 0-7)← addr 0-7 (PC 11) ← DBF	Call designated Subroutine.	2	2				
RET	(SP) ← (SP) - 1 (PC) ← ((SP))	Return from Subroutine without restoring Program Status Word.	2	1				
RETR	(SP) ← (SP) - 1 (PC) ← ((SP)) (PSW 4-7) ← ((SP))	Return from Subroutine restoring Program Status Word.	2	1	•	•		
		FLAGS						
CPL C	(C) ← NOT (C)	Complement Content of carry bit.	1	1	•			
CPL F0	(F0) ← NOT (F0)	Complement Content of Flag F0.	1	1			•	
CPL F1	(F1) ← NOT (F1)	Complement Content of Flag F1.	1	1				•
CLR C	(C) ← 0	Clear content of carry bit to 0.	1	1	•			
CLR F0	(F0) ← 0.	Clear content of Flag 0 to 0.	1	1			•	
CLR F1	(F1) ← 0	Clear content of Flag 1 to 0.	1	1				•
		MISCELLANEOUS						
NOP		No operation	1	1				

Symbol Definitions

SYMBOL	DESCRIPTION
A	The Accumulator
AC	The Auxiliary Carry Flag
addr	Program Memory Address (12 bits)
b	Bit Designator (b = 0 - 7)
BS	The Bank Switch
BUS	The BUS Port
C	Carry Flag
CLK	Clock Signal
CNT	Event Counter
D	Nibble Designator (4 bits)
data	Number or Expression (8 bits)
DBF	Memory Bank Flip-Flop
F_0 F_1	Flags 0, 1
I	Interrupt
P	"In-Page" Operation Designator

SYMBOL	DESCRIPTION
P	Port Designator (p = 1, 2 or 4 - 7)
PSW	Program Status Word
r	Register Designator (r = 0, 1 or 0 - 7)
SP	Stack Pointer
T	Timer
TF	Timer Flag
T_0 T_1	Testable Flags 0, 1
X	External RAM
#	Prefix for Immediate Data
@	Prefix for Indirect Address
$	Program Counter's Current Value
(x)	Contents of External RAM Location
((xx))	Contents of Memory Location Addressed by the Contents of External RAM Location.
←	Replaced By

TYPICAL APPLICATIONS

Figure 6 shows a typical way to use the 48-Series Microcomputers in a stand-alone system.

- Crystal used is:
  - Series resonant
  - AT cut
  - 1 to 6 MHz or 4 to 11 MHz

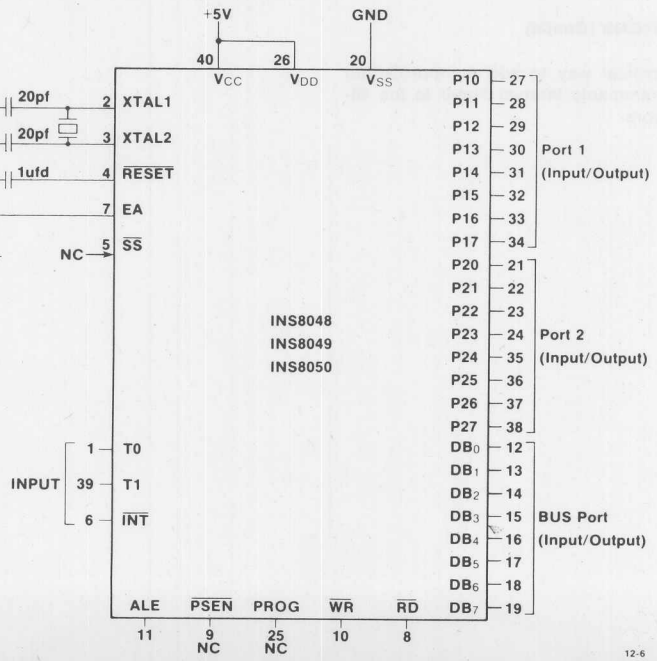


FIGURE 6. Stand-alone INS8048/49/50

TYPICAL APPLICATIONS (Cont'd)

Figure 7 shows a typical remote data acquisition system with an INS8250 Programmable Asynchronous Communication System which can receive commands or update information from a supervisory computer. The figure also shows an INS8294 CMOS DVM that receives data at V_{IN} and displays the data on the 7-segment local display unit. Data are transferred from the INS8294 to the INS8049 via National's MICROBUS™.

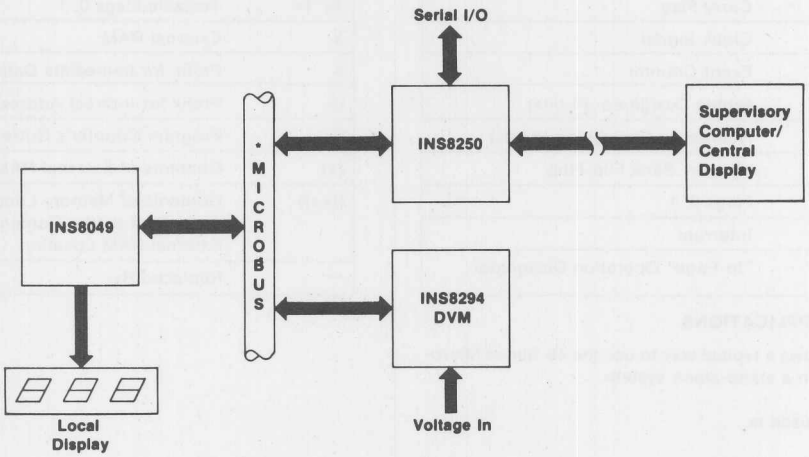


FIGURE 7. Data Acquisition System

12-7

TYPICAL APPLICATIONS (Cont'd)

Figure 8 shows a typical way to add a Input/Output Expander and Programmable Interval Timer to the 48-Series Microprocessors.

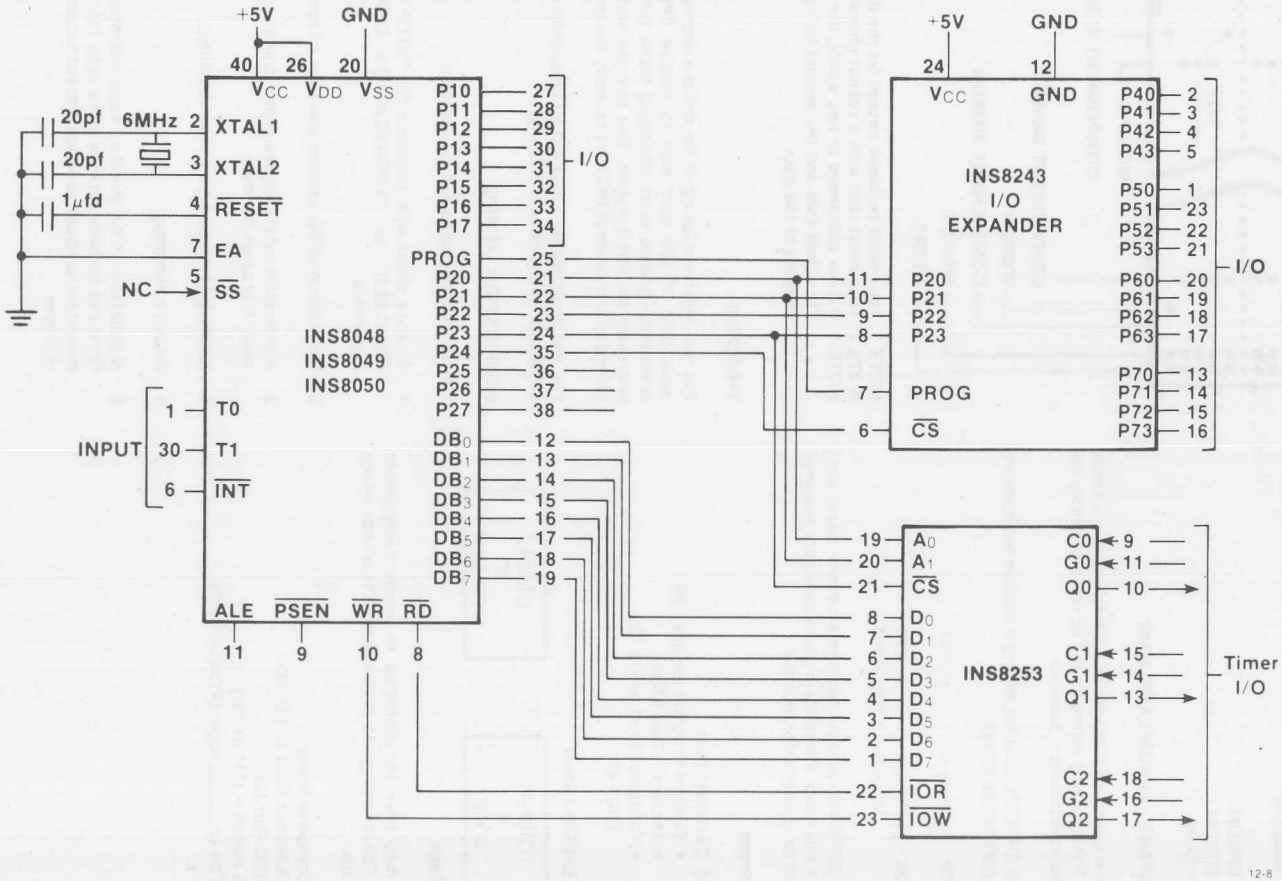


FIGURE 8. INS8253 Interval Timer

APPROVED FORMATS FOR CUSTOM PROGRAMMED PARTS

INPUT MEDIUM:

2716 EPROM
2708 EPROM
PAPER TAPE

IMPORTANT - EPROM LABELLING

Only one customer program may be included in a single order. The following method must be used to identify the EPROMs comprising a program.

a) The EPROMs used for storing a custom program are designated as shown:

2716:	Block A	0-2047
2708:	Block A	0-1023
	Block B	1024-2047

b) All EPROMs must be labelled (stickers, paint, etc.) with this block designation plus a customer assigned print or identification number.

Example:

1) Customer Data
   - Custom Program Length - 2K
   - Medium - Two 2708's
   - Customer Print or I.D. No. C123-45

2) EPROM Labels

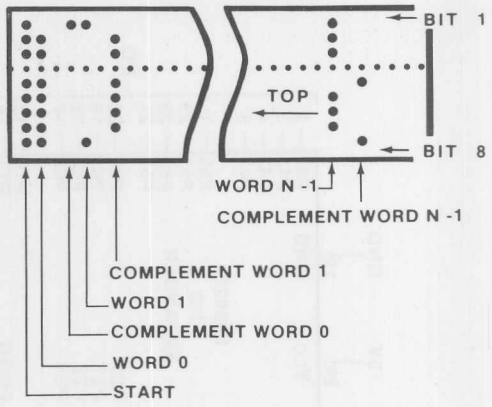
C123-45 A 0-1023	C123-45 B 1024-2047

Paper Tape

Tapes may only be submitted in binary complement format. The following information should be written on the paper tape.

Company Name
Customer Print or I.D. No.
NSC Part No.
A Punch = ("1" or "0")
This is _____ logic (POS or NEG)

BINARY COMPLEMENT FORMAT



NOTE 1: Tape must be blank except for the data words.
NOTE 2: Tape must start with a rubout character.
NOTE 3: Data is comprised of two words, the first being the actual data and the second being the complement of the data.

Verification

You will receive a listing of the options ordered and the input data. If you also wish to receive EPROMs for verification, please send additional blank EPROMs as necessary for this purpose. You can use software (the listing) or hardware (EPROMs) to verify the program.

You will be asked for a GO/NO GO response within one week after you receive the listing.

VERIFICATION LISTING

The verification listing has six sections:

1. A cover sheet with provision for "STOP, DO NOT PROCEED" or "VERIFICATION CERTIFIED" signatures.
2. Description of the options you have chosen.
3. A description of the log designations and assumptions used to process the data.
4. A listing of the data you have submitted.
5. An error summary.
6. A definition of the standard logic definitions for the ROM and the reduced form of the data. This list shows the output word corresponding to each address coded in binary.

Ordering Information for Custom Programmed Parts

The following information must be submitted with each customer microcomputer program. An order will not be processed unless it is accompanied by this information. This form acts as a Traveler from Customer through Customer Service to ROM programming. Please retain a copy of this form to compare against the verification listing. The form will be sent back to the customer by Customer Service.

	National Microcomputer Part Number
	ROM Letter Code (National Use Only)
Name	Date
Address	Customer Print or I.D. No.
City State Zip	Purchase Order No.
Telephone ( )	Name of person National can contact (Print)
Authorized Signature	Date

OPTIONS

1. Device Type (Circle One)

   8048-6 (6 MHz) 8049-6 8050-6

   8048-11 (11 MHz) 8049-11

   A. Input Medium (Circle One)

   48 Series Part: 8048 8748

   EPROM: 2708 2716 2758A 2732 2758B

   Total number of EPROMs: _____

   B. Verification Medium: The user will be sent one or more of the following media to verify National's reception of valid date. If any EPROM is selected, blank EPROMs must be submitted with this form.

   Circle One or More

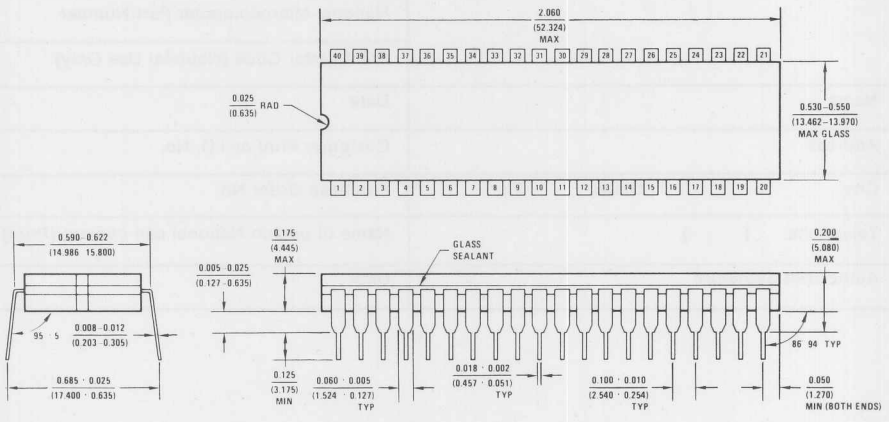
   Listing: Binary MPS Format Hexadecimal Format

   EPROM: 2708 2716 2758A 2732 2758B 8748

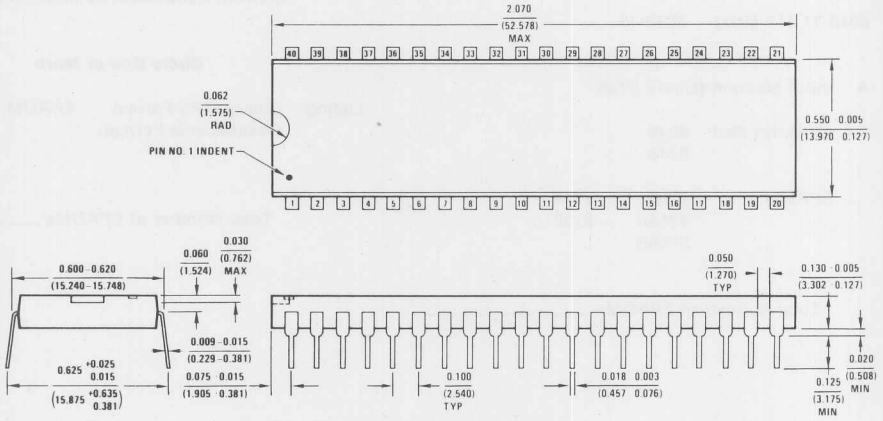
   Total Number of EPROMs: _____

Physical Dimensions

inches (millimeters)



Ceramic Dual-in-Line Package (J)
Order Number INS8048J



Plastic Dual-in-Line Package (N)
Order Number INS8048N

National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, California 95051
Tel: (408) 737-5000
TWX: (910) 339-9240

National Semiconductor GmbH
Eisenheimerstrasse 61/II
8000 Munchen 21
West Germany
Tel: (089) 576091
Telex: 05 22772

NS International Inc., Japan
Miyake Building
1-9 Yotsuya, Shinjuku-ku 160
Tokyo, Japan
Tel: (03) 355-3711
TWX: 232 2015 NSCJ J

National Semiconductor (Hong Kong) Ltd.
8th Floor
Cheung Kong Electronic Bldg
4 Hing Yip Street
Kwun Tong
Kowloon, Hong Kong
Tel: 3 899235
Telex: 73866 NSEHK HX
Cable: NATSEMI

NS Electronics Do Brasil
Avda Brigadeiro Faria Lima 844
11 Andar Conjunto 1104
Jardim Paulistano
Sao Paulo, Brasil
Telex:
1121008 CABINE SAO PAULO

NS Electronics Pty. Ltd.
Cnr. Stud Rd. & Mtn. Highway
Bayswater, Victoria 3153
Australia
Tel: 03 729 6333
Telex: 32096

Analog I/O Components

National Semiconductor

ADC0801, ADC0802, ADC0803, ADC0804 8-Bit μP Compatible A/D Converters

General Description

The ADC0801, ADC0802, ADC0803, ADC0804 are CMOS 8-bit, successive approximation A/D converters which use a modified potentiometric ladder—similar to the 256R products. They are designed to meet the NSC MICROBUS™ standard to allow operation with the 8080A control bus, and TRI-STATE® output latches directly drive the data bus. These A/Ds appear like memory locations or I/O ports to the microprocessor and no interfacing logic is needed.

A new differential analog voltage input allows increasing the common-mode rejection and offsetting the analog zero input voltage value. In addition, the voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 8 bits of resolution.

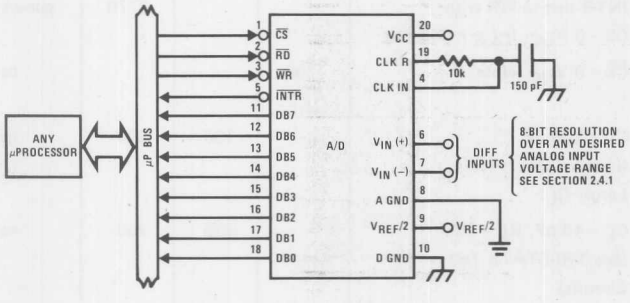
Features

- MICROBUS (8080A) compatible—no interfacing logic needed
- Easy interface to all microprocessors, or operates "stand alone"
- Differential analog voltage inputs
- Logic inputs and outputs meet T^2L voltage level specifications
- Works with 2.5V (LM336) voltage reference
- On-chip clock generator
- 0V to 5V analog input voltage range with single 5V supply
- No zero adjust required
- 0.3" standard width 20-pin DIP package

Key Specifications

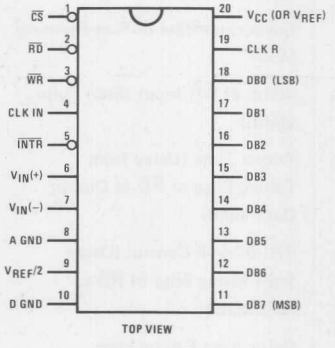
■ Resolution	8 bits
■ Total error	±1/4 LSB, ±1/2 LSB and ±1 LSB
■ Conversion time	100 μs
■ Access time	135 ns
■ Single supply	5 V_{DC}
■ Operates ratiometrically or with 5 V_{DC}, 2.5 V_{DC}, or analog span adjusted voltage reference	

Typical Application



Connection Diagram

ADC 080X
Dual-In-Line Package



TOP VIEW

Absolute Maximum Ratings (Notes 1 and 2)

Supply Voltage (V_{CC}) (Note 3)	6.5V
Voltage at Any Input	−0.3V to (V_{CC} +0.3V)
Storage Temperature Range	−65°C to +150°C
Package Dissipation at T_A = 25°C	875 mW
Lead Temperature (Soldering, 10 seconds)	300°C

Operating Ratings (Notes 1 and 2)

Temperature Range (Note 1)	$T_{MIN} \leq T_A \leq T_{MAX}$
ADC0801/02/03 LD	−55°C ≤ T_A ≤ +125°C
ADC0801/02/03/04 LCD	−40°C ≤ T_A ≤ +85°C
ADC0801/02/03/04 LCN	0°C ≤ T_A ≤ 70°C
Range of V_{CC} (Note 1)	4.5 V_{DC} to 6.3 V_{DC}

Electrical Characteristics

Converter Specifications:

V_{CC} = 5 V_{DC}, $V_{REF}/2$ = 2.500 V_{DC}, $T_{MIN} \leq T_A \leq T_{MAX}$ and f_{CLK} = 640 kHz unless otherwise stated.

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
ADC0801:					
Total Adjusted Error (Note 8)	With Full-Scale Adj.			±1/4	LSB
ADC0802:					
Total Unadjusted Error (Note 8)	Completely Unadjusted			±1/2	LSB
ADC0803:					
Total Adjusted Error (Note 8)	With Full-Scale Adj.			±1/2	LSB
ADC0804:					
Total Unadjusted Error (Note 8)	Completely Unadjusted			±1	LSB
$V_{REF}/2$ Input Resistance	Input Resistance at Pin 9	1.0	1.3		kΩ
Analog Input Voltage Range	(Note 4) V(+) or V(−)	Gnd−0.05		V_{CC}+0.05	V_{DC}
DC Common-Mode Rejection	Over Analog Input Voltage Range		±1/16	±1/8	LSB
Power Supply Sensitivity	V_{CC} = 5 V_{DC} ±10% Over Allowed V_{IN}(+) and V_{IN}(−) Voltage Range (Note 4)		±1/16	±1/8	LSB

Electrical Characteristics

Timing Specifications: V_{CC} = 5 V_{DC} and T_A = 25°C unless otherwise noted.

	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
f_{CLK}	Clock Frequency	V_{CC} = 6V, (Note 5)	100	640	1280	kHz
		V_{CC} = 5V	100	640	800	kHz
T_c	Conversion Time	(Note 6)	66		73	$1/f_{CLK}$
CR	Conversion Rate In Free-Running Mode	$\overline{INTR}$ tied to $\overline{WR}$ with $\overline{CS}$ = 0 V_{DC}, f_{CLK} = 640 kHz			8770	conv/s
$t_{W(\overline{WR})L}$	Width of $\overline{WR}$ Input (Start Pulse Width)	$\overline{CS}$ = 0 V_{DC} (Note 7)	100			ns
t_{ACC}	Access Time (Delay from Falling Edge of $\overline{RD}$ to Output Data Valid)	C_L = 100 pF (Use Bus Driver IC for Larger C_L)		135	200	ns
t_{1H}, t_{0H}	TRI-STATE Control (Delay from Rising Edge of $\overline{RD}$ to Hi-Z State)	C_L = 10 pF, R_L = 10k (See TRI-STATE Test Circuits)		125	250	ns
t_{WI}	Delay from Falling Edge of $\overline{WR}$ to Reset of $\overline{INTR}$			300	450	ns
C_{IN}	Input Capacitance of Logic Control Inputs			5	7.5	pF
C_{OUT}	TRI-STATE Output Capacitance (Data Buffers)			5	7.5	pF

Electrical Characteristics

Digital Levels and DC Specifications:

V_{CC} = 5 V_{DC} and $T_{MIN} \leq T_A \leq T_{MAX}$, unless otherwise noted.

PARAMETER		CONDITIONS	MIN	TYP	MAX	UNITS
CONTROL INPUTS [Note: CLK IN (pin 4) is the input of a Schmitt trigger circuit and is therefore specified separately]						
V_{IN} (1)	Logical "1" Input Voltage (Except Pin 4 CLK IN)	V_{CC} = 5.25 V_{DC}	2.0		15	V_{DC}
V_{IN} (0)	Logical "0" Input Voltage (Except Pin 4 CLK IN)	V_{CC} = 4.75 V_{DC}			0.8	V_{DC}
V_T+	CLK IN (Pin 4) Positive Going Threshold Voltage		2.7	3.1	3.5	V_{DC}
V_T-	CLK IN (Pin 4) Negative Going Threshold Voltage		1.5	1.8	2.1	V_{DC}
V_H	CLK IN (Pin 4) Hysteresis $(V_T+) - (V_T-)$		0.6	1.3	2.0	V_{DC}
I_{IN} (1)	Logical "1" Input Current (All Inputs)	V_{IN} = 5 V_{DC}		0.005	1	μA_{DC}
I_{IN} (0)	Logical "0" Input Current (All Inputs)	V_{IN} = 0 V_{DC}	−1	−0.005		μA_{DC}
I_{CC}	Supply Current (Includes Ladder Current)	f_{CLK} = 640 kHz, T_A = 25°C and $\overline{CS}$ = "1"		1.3	2.5	mA
DATA OUTPUTS AND $\overline{INTR}$						
V_{OUT} (0)	Logical "0" Output Voltage	I_O = 1.6 mA V_{CC} = 4.75 V_{DC}			0.4	V_{DC}
V_{OUT} (1)	Logical "1" Output Voltage	I_O = −360 μA V_{CC} = 4.75 V_{DC}	2.4			V_{DC}
I_{OUT}	TRI-STATE Disabled Output Leakage (All Data Buffers)	V_{OUT} = 0 V_{DC}	−3			μA_{DC}
		V_{OUT} = 5 V_{DC}			3	μA_{DC}
	Output Short Circuit Current	T_A = 25°C				
I_{SOURCE}		V_{OUT} Short to Gnd	4.5	6		mA_{DC}
I_{SINK}		V_{OUT} Short to V_{CC}	9.0	16		mA_{DC}

Note 1: Absolute maximum ratings are those values beyond which the life of the device may be impaired.

Note 2: All voltages are measured with respect to Gnd, unless otherwise specified. The separate A Gnd point should always be wired to the D Gnd.

Note 3: A zener diode exists, internally, from V_{CC} to Gnd and has a typical breakdown voltage of 7 V_{DC}.

Note 4: For $V_{IN}(-) \geq V_{IN}(+)$ the digital output code will be 0000 0000. Two on-chip diodes are tied to each analog input (see block diagram) which will forward conduct for analog input voltages one diode drop below ground or one diode drop greater than the V_{CC} supply. Be careful, during testing at low V_{CC} levels (4.5V), as high level analog inputs (5V) can cause this input diode to conduct—especially at elevated temperatures, and cause errors for analog inputs near full-scale. The spec allows 50 mV forward bias of either diode. This means that as long as the analog V_{IN} does not exceed the supply voltage by more than 50 mV, the output code will be correct. To achieve an absolute 0 V_{DC} to 5 V_{DC} input voltage range will therefore require a minimum supply voltage of 4.950 V_{DC} over temperature variations, initial tolerance and loading.

Note 5: With V_{CC} = 6V, the digital logic interfaces are no longer TTL compatible.

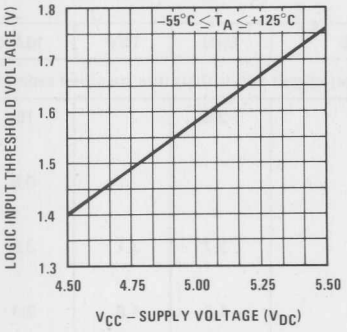
Note 6: With an asynchronous start pulse, up to 8 clock periods may be required before the internal clock phases are proper to start the conversion process.

Note 7: The $\overline{CS}$ input is assumed to bracket the $\overline{WR}$ strobe input and therefore timing is dependent on the $\overline{WR}$ pulse width. An arbitrarily wide pulse width will hold the converter in a reset mode and the start of conversion is initiated by the low to high transition of the $\overline{WR}$ pulse (see timing diagrams).

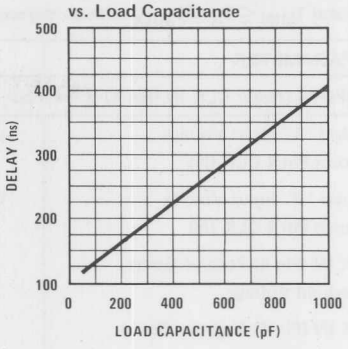
Note 8: None of these A/Ds requires a zero adjust. However, if an all zero code is desired for an analog input other than 0.0V, or if a narrow full-scale span exists (for example: 0.5V to 4.0V full-scale) the $V_{IN}(-)$ input can be adjusted to achieve this. See section 2.5 and *Figure 19*.

Typical Performance Characteristics

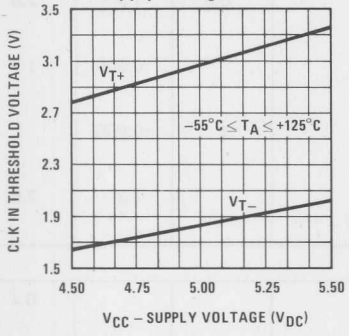
Logic Input Threshold Voltage vs. Supply Voltage



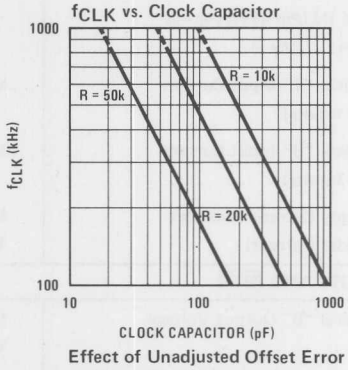
Delay From Falling Edge of $\overline{RD}$ to Output Data Valid vs. Load Capacitance



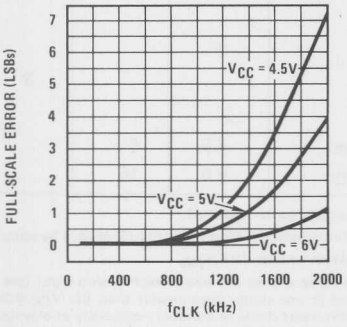
CLK IN Schmitt Trip Levels vs. Supply Voltage



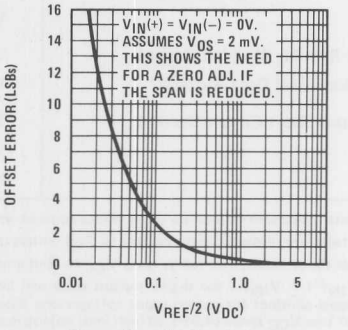
f_{CLK} vs. Clock Capacitor



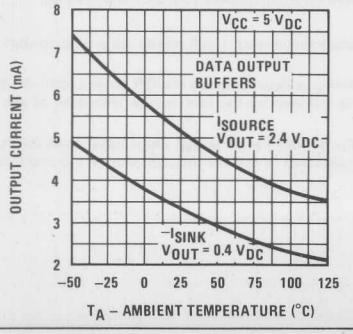
Full-Scale Error vs f_{CLK}



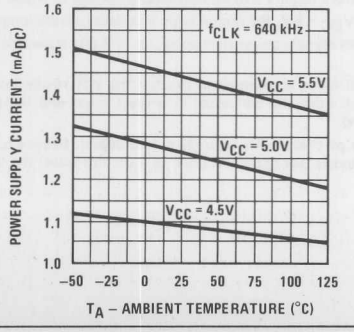
Effect of Unadjusted Offset Error vs. $V_{REF}/2$ Voltage



Output Current vs Temperature

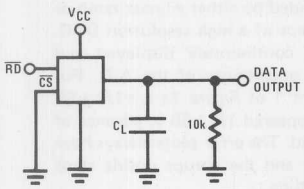


Power Supply Current vs. Temperature

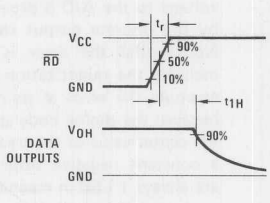


TRI-STATE® Test Circuits and Waveforms

t_{1H}

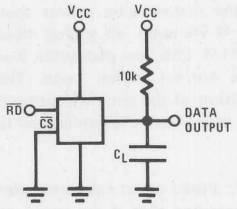


t_{1H}, C_L = 10 pF

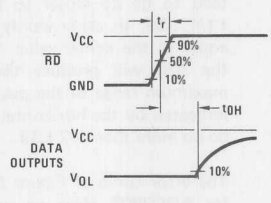


t_r = 20 ns

t_{0H}

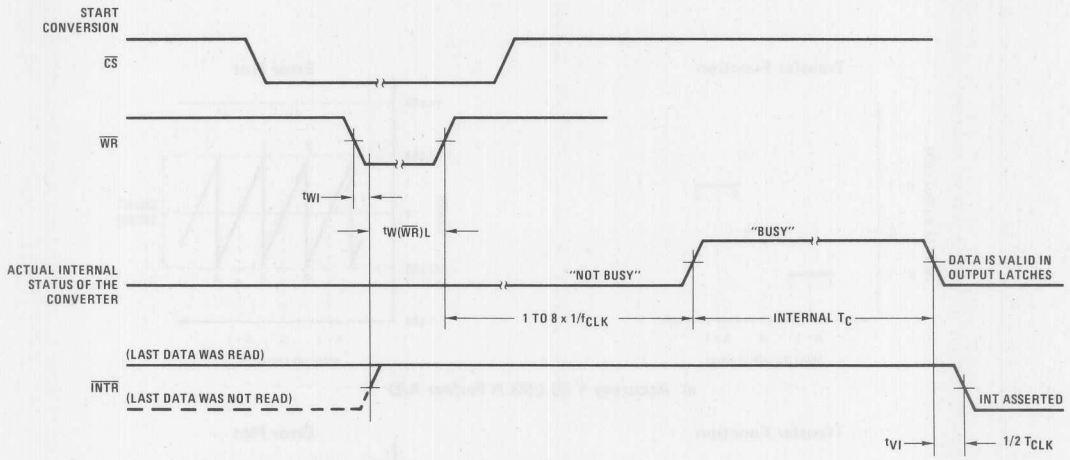


t_{0H}, C_L = 10 pF

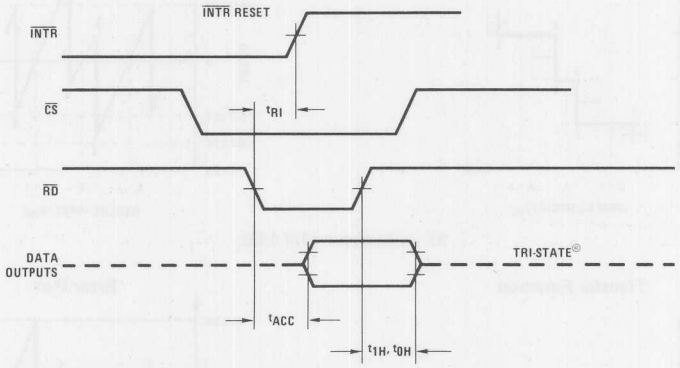


t_r = 20 ns

Timing Diagrams



Output Enable and Reset $\overline{INTR}$



Note: All timing is measured from the 50% voltage points.

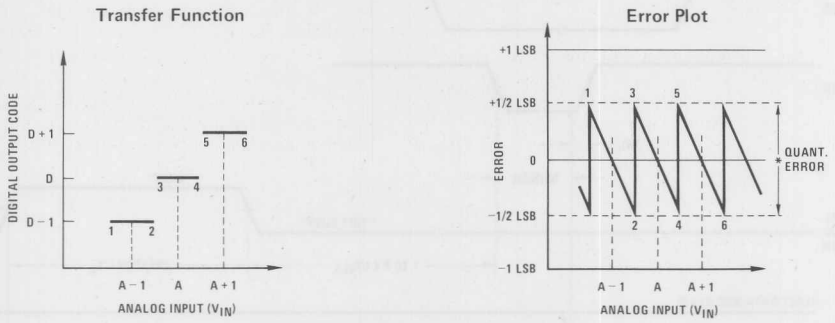
1.0 UNDERSTANDING A/D ERROR SPECS

A perfect A/D transfer characteristic (staircase waveform) is shown in *Figure 1a.* The horizontal scale is analog input voltage and the particular points labeled are in steps of 1 LSB (19.53 mV with 2.5V tied to the $V_{REF}/2$ pin). The digital output codes which correspond to these inputs are shown as D−1, D, and D+1. For the perfect A/D, not only will center-value (A−1, A, A+1, . . .) analog inputs produce the correct output digital codes, but also each riser (the transitions between adjacent output codes) will be located ±1/2 LSB away from each center-value. As shown, the risers are ideal and have no width. Correct digital output codes will be provided for a range of analog input voltages which extend ±1/2 LSB from the ideal center-values. Each tread (the range of analog input voltage which provides the same digital output code) is therefore 1 LSB wide.

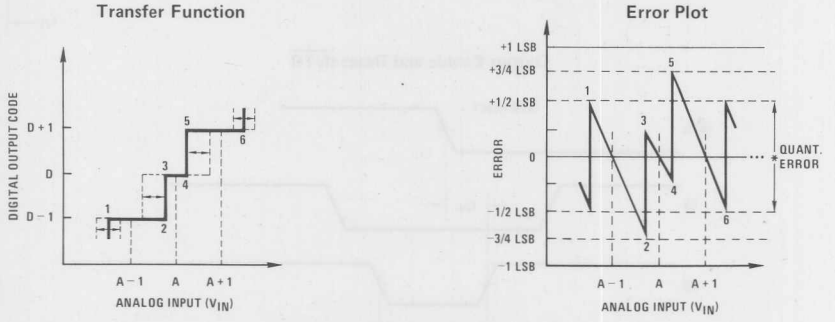
Figure 1b shows worst case error plot for ADC0801. All center-valued inputs are guaranteed to produce the correct output codes and the adjacent risers are guaranteed to be no closer to the center-value points than ±1/4 LSB. In other words, if we apply an analog input equal to the center-value ±1/4 LSB, *we guarantee* that the A/D will produce the correct digital code. The maximum range of the position of the code transition is indicated by the horizontal arrow and it is guaranteed to be no more than 1/2 LSB.

The error curve of *Figure 1c* shows worst case error plot for ADC0802. Here we guarantee that if we apply an analog input equal to the LSB analog voltage center-value the A/D will produce the correct digital code.

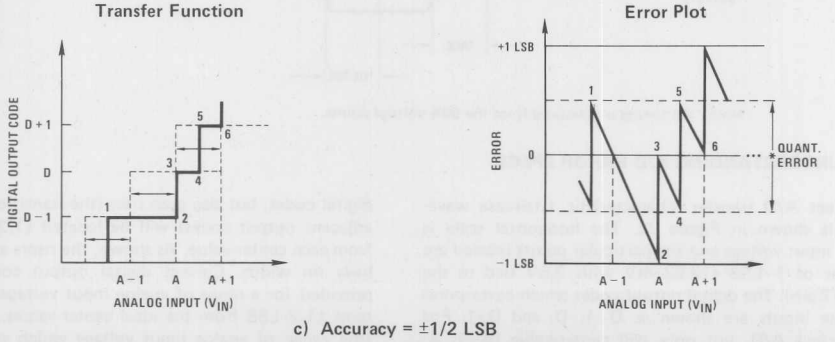
Next to each transfer function is shown the corresponding error plot. Many people may be more familiar with error plots than transfer functions. The analog input voltage to the A/D is provided by either a linear ramp or by the discrete output steps of a high resolution DAC. Notice that the error is continuously displayed and includes the quantization uncertainty of the A/D. For example the error at point 1 of *Figure 1a* is +1/2 LSB because the digital code appeared 1/2 LSB in advance of the center-value of the tread. The error plots always have a constant negative slope and the abrupt upside steps are always 1 LSB in magnitude.



a) Accuracy = ±0 LSB A Perfect A/D



b) Accuracy = ±1/4 LSB



c) Accuracy = ±1/2 LSB

FIGURE 1. Clarifying the Error Specs of an A/D Converter

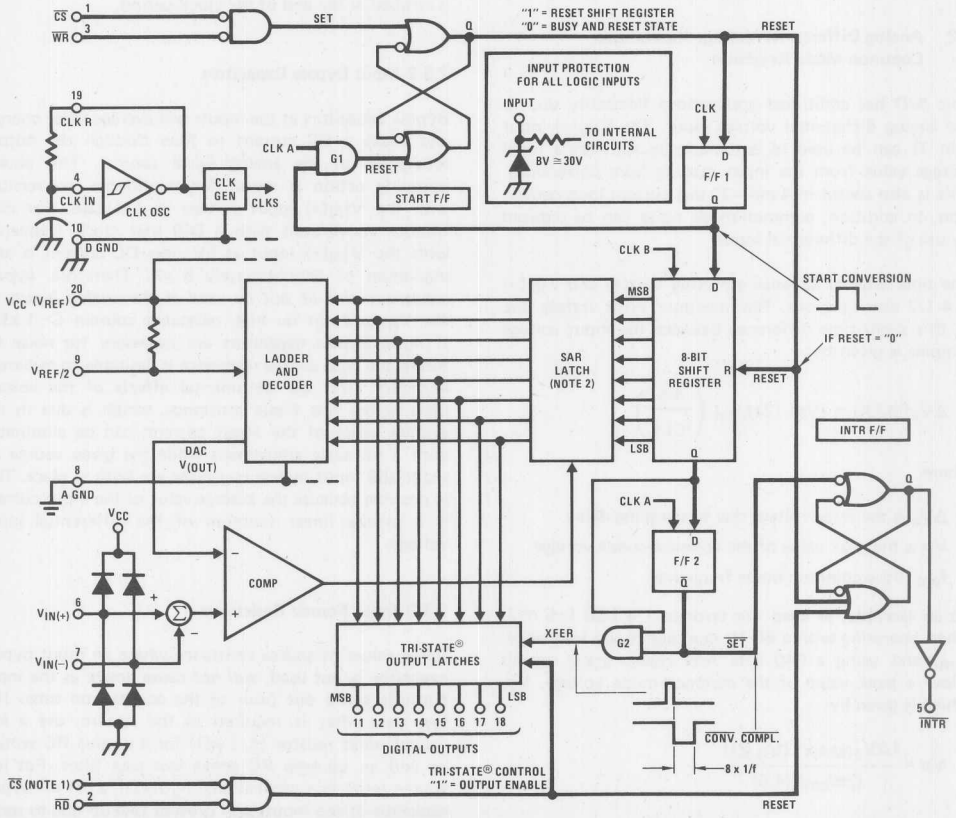
2.0 FUNCTIONAL DESCRIPTION

The ADC0801 series contains a circuit equivalent of the 256R network. Analog switches are sequenced by successive approximation logic to match the analog difference input voltage $[V_{IN}(+) - V_{IN}(-)]$ to a corresponding tap on the R network. The most significant bit is tested first and after 8 comparisons (64 clock cycles) a digital 8-bit binary code (1111 1111 = full-scale) is transferred to an output latch and then an interrupt is asserted ($\overline{INTR}$ makes a high-to-low transition). The device may be operated in the free-running mode by connecting $\overline{INTR}$ to the $\overline{WR}$ input with $\overline{CS}$ = 0. To insure start-up under all possible conditions, an external $\overline{WR}$ pulse is required during the first power-up cycle. A conversion in process can be interrupted by issuing a second start command.

On the high-to-low transition of the $\overline{WR}$ input the internal SAR latches and the shift register stages are reset. As long as the $\overline{CS}$ input and $\overline{WR}$ input remain low, the A/D will remain in a reset state. *Conversion will start from 1 to 8 clock periods after at least one of these inputs makes a low-to-high transition.*

A functional diagram of the A/D converter is shown in *Figure 2.* All of the package pinouts are shown and the major logic control paths are drawn in heavier weight lines.

The converter is started by having $\overline{CS}$ and $\overline{WR}$ simultaneously low. This sets the start flip-flop (F/F) and the resulting "1" level resets the 8-bit shift register, resets the Interrupt ($\overline{INTR}$) F/F and inputs a "1" to the D flop, F/F1, which is at the input end of the 8-bit shift register. Internal clock signals then transfer this "1" to the Q output of F/F1. The AND gate, G1, combines this "1" output with a clock signal to provide a reset signal to the start F/F. If the set signal is no longer present (either $\overline{WR}$ or $\overline{CS}$ is a "1") the start F/F is reset and the 8-bit shift register then can have the "1" clocked in, which starts the conversion process. If the set signal were to still be present, this reset pulse would have no effect (both outputs of the start F/F would momentarily be at a "1" level) and the 8-bit shift register would continue to be held in the reset mode. This logic therefore allows for wide $\overline{CS}$ and $\overline{WR}$ signals and the converter will start after at least one of these signals returns high and the internal clocks again provide a reset signal for the start F/F.



Note 1: $\overline{CS}$ shown twice for clarity.

Note 2: SAR = Successive Approximation Register.

FIGURE 2. Block Diagram

After the "1" is clocked through the 8-bit shift register (which completes the SAR search) it appears as the input to the D flop, F/F 2. As soon as this "1" is output from the shift register, the AND gate, G2, causes the new digital word to transfer to the TRI-STATE output latches. When F/F 2 is subsequently clocked, the $\overline{Q}$ output makes a high-to-low transition which causes the INTR F/F to set. An inverting buffer then supplies the INTR output signal.

When data is to be read, the combination of both $\overline{CS}$ and $\overline{RD}$ being low will cause the INTR F/F to be reset and the TRI-STATE output latches will be enabled to provide the 8-bit digital outputs.

2.1 Digital Control Inputs

The digital control inputs ($\overline{CS}$, $\overline{RD}$, and $\overline{WR}$) meet standard T^2L logic voltage levels. These signals have been renamed when compared to the standard A/D Start and Output Enable labels. In addition, these inputs are active low to allow an easy interface to microprocessor control busses. For non-microprocessor based applications, the $\overline{CS}$ input (pin 1) can be grounded and the standard A/D Start function is obtained by an active low pulse applied at the $\overline{WR}$ input (pin 3) and the Output Enable function is caused by an active low pulse at the $\overline{RD}$ input (pin 2).

2.2 Analog Differential Voltage Inputs and Common-Mode Rejection

This A/D has additional applications flexibility due to the analog differential voltage input. The $V_{IN}(-)$ input (pin 7) can be used to automatically subtract a fixed voltage value from the input reading (tare correction). This is also useful in 4 mA–20 mA current loop conversion. In addition, common-mode noise can be reduced by use of the differential input.

The time interval between sampling $V_{IN}(+)$ and $V_{IN}(-)$ is 4-1/2 clock periods. The maximum error voltage due to this slight time difference between the input voltage samples is given by:

$$\Delta V_e(MAX) = (V_P)\ (2\pi f_{cm})\left(\frac{4.5}{f_{CLK}}\right)$$

where:

ΔV_e is the error voltage due to sampling delay

V_P is the peak value of the common-mode voltage

f_{cm} is the common-mode frequency

As an example, to keep this error to 1/4 LSB (~5 mV) when operating with a 60 Hz common-mode frequency, f_{cm}, and using a 640 kHz A/D clock, f_{CLK}, would allow a peak value of the common-mode voltage, V_P, which is given by:

$$V_P = \frac{[\Delta V_e(MAX)\ (f_{CLK})]}{(2\pi f_{cm})\ (4.5)}$$

or

$$V_P = \frac{(5 \times 10^{-3})\ (640 \times 10^{3})}{(6.28)\ (60)\ (4.5)}$$

which gives

$V_P \cong 1.9V$.

The allowed range of analog input voltages usually places more severe restrictions on input common-mode noise levels.

An analog input voltage with a reduced span and a relatively large zero offset can be easily handled by making use of the differential input (see section 2.4 Reference Voltage Flexibility).

2.3 Analog Inputs

2.3.1 Input Current

Due to the internal switching action, displacement currents will flow at the analog inputs. This is due to on-chip stray capacitance to ground. The voltage on this capacitance is switched and will result in currents entering the $V_{IN}(+)$ input and leaving the $V_{IN}(-)$ input which will depend on the analog differential input voltage levels. These current transients occur at the leading edge of the internal clocks. They rapidly decay and *do not cause errors* as the on-chip comparator is strobed at the end of the clock period.

2.3.2 Input Bypass Capacitors

Bypass capacitors at the inputs will average these charges and cause a DC current to flow through the output resistances of the analog signal sources. This charge pumping action is worse for continuous conversions with the $V_{IN}(+)$ input voltage at full-scale. For continuous conversions with a 640 kHz clock frequency with the $V_{IN}(+)$ input at 5V, this DC current is at a maximum of approximately 5 μA. Therefore, *bypass capacitors should not be used at the analog inputs or the $V_{REF}/2$ pin* for high resistance sources ($> 1\ k\Omega$). If input bypass capacitors are necessary for noise filtering and high source resistance is desirable to minimize capacitor size, the detrimental effects of the voltage drop across this input resistance, which is due to the average value of the input current, can be eliminated with a full-scale adjustment while the given source resistor and input bypass capacitor are both in place. This is possible because the average value of the input current is a precise linear function of the differential input voltage.

2.3.3 Input Source Resistance

Large values of source resistance where an input bypass capacitor is not used, *will not cause errors* as the input currents settle out prior to the comparison time. If a low pass filter is required in the system, use a low valued series resistor ($\leq 1\ k\Omega$) for a passive RC section or add an op amp RC active low pass filter. For low source resistance applications, ($\leq 1\ k\Omega$), a 0.1 μF bypass capacitor at the inputs will prevent pickup due to series lead inductance of a long wire. A 100Ω series resistor can be used to isolate this capacitor—both the R and C are placed outside the feedback loop—from the output of an op amp, if used.

2.3.4 Noise

The leads to the analog inputs (pins 6 and 7) should be kept as short as possible to minimize input noise coupling. Both noise and undesired digital clock coupling to these inputs can cause system errors. The source resistance for these inputs should, in general, be kept below 5 kΩ. Larger values of source resistance can cause undesired system noise pickup. Input bypass capacitors, placed from the analog inputs to ground, will eliminate system noise pickup but can create analog scale errors as these capacitors will average the transient input switching currents of the A/D (see section 2.3.3). This scale error depends on both a large source resistance and the use of an input bypass capacitor. This error can be eliminated by doing a full-scale adjustment of the A/D (adjust $V_{REF}/2$ for a proper full-scale reading—see section 2.5.2 on Full-Scale Adjustment) with the source resistance and input bypass capacitor in place.

2.4 Reference Voltage

2.4.1 Span Adjust

For maximum applications flexibility, these A/Ds have been designed to accommodate a 5 V_{DC}, 2.5 V_{DC} or an adjusted voltage reference. This has been achieved in the design of the IC as shown in *Figure 3*.

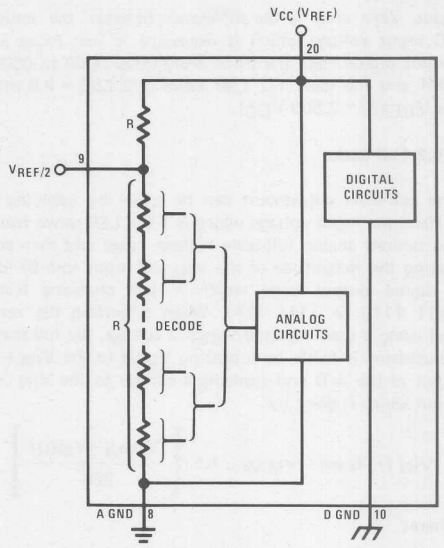


FIGURE 3. The $V_{REFERENCE}$ Design on the IC

Notice that the reference voltage for the IC is either 1/2 of the voltage which is applied to the V_{CC} supply pin, or is equal to the voltage which is externally forced at the $V_{REF}/2$ pin. This allows for a ratiometric voltage reference using the V_{CC} supply, a 5 V_{DC} reference voltage can be used for the V_{CC} supply or a voltage less than 2.5 V_{DC} can be applied to the $V_{REF}/2$ input for increased application flexibility. The internal gain to the $V_{REF}/2$ input is 2 to allow this factor of 2 reduction in the $V_{REF}/2$ voltage.

An example of the use of an adjusted reference voltage is to accommodate a reduced span—or dynamic voltage range of the analog input voltage. If the analog input voltage were to range from 0.5 V_{DC} to 3.5 V_{DC}, instead of 0V to 5 V_{DC}, the span would be 3V. With 0.5 V_{DC} applied to the V_{IN} (−) pin to absorb the offset, the reference voltage can be made equal to 1/2 of the 3V span or 1.5 V_{DC}. The A/D now will encode the V_{IN} (+) signal from 0.5V to 3.5V with the 0.5V input corresponding to zero and the 3.5 V_{DC} input corresponding to full-scale. The full 8 bits of resolution are therefore applied over this reduced analog input voltage range.

2.4.2 Reference Accuracy Requirements

The converter can be operated in a ratiometric mode or an absolute mode. In ratiometric converter applications, the magnitude of the reference voltage is a factor in both the output of the source transducer and the output of the A/D converter and therefore cancels out in the final digital output code. In absolute conversion applications, both the initial value and the temperature stability of the reference voltage are important accuracy factors in the operation of the A/D converter. For $V_{REF}/2$ voltages of 2.5 V_{DC} nominal value, initial errors of ±10 mV_{DC} will cause conversion errors of ±1 LSB due to the gain of 2 of the $V_{REF}/2$ input. In reduced span applications, the initial value and the stability of the $V_{REF}/2$ input voltage become even more important. For example, if the span is reduced to 2.5V, the analog input LSB voltage value is correspondingly reduced from 20 mV (5V span) to 10 mV and 1 LSB at the $V_{REF}/2$ input becomes 5 mV. As can be seen, this reduces the allowed initial tolerance of the reference voltage and requires correspondingly less absolute change with temperature variations. Note that spans smaller than 2.5V place even tighter requirements on the initial accuracy and stability of the reference source.

In general, the magnitude of the reference voltage will require an initial adjustment. Errors due to an improper value of reference voltage appear as full-scale errors in the A/D transfer function. IC voltage regulators may be used for references if the ambient temperature changes are not excessive. The LM336B 2.5V IC reference diode (from National Semiconductor) is available which operates with a 5V input voltage and has a temperature stability of 1.8 mV typ (6 mV max) over $0^{\circ}C \leq T_A \leq +70^{\circ}C$. Other temperature range parts are also available.

2.5 Errors

2.5.1 Zero Error

The zero of the A/D does not require adjustment. If the minimum analog input voltage value, $V_{IN(MIN)}$, is not ground, a zero offset can be done. The converter can be made to output 0000 0000 digital code for this minimum input voltage by biasing the A/D V_{IN} (−) input at this $V_{IN(MIN)}$ value (see Applications section). This utilizes the differential mode operation of the A/D.

The zero error of the A/D converter relates to the location of the first riser of the transfer function and can be measured by grounding the V (−) input and applying a small magnitude positive voltage to the V (+)

input. Zero error is the difference between the actual DC input voltage which is necessary to just cause an output digital code transition from 0000 0000 to 0000 0001 and the ideal 1/2 LSB value (1/2 LSB = 9.8 mV for $V_{REF}/2 = 2.500\ V_{DC}$).

2.5.2 Full-Scale

The full-scale adjustment can be made by applying a differential input voltage which is 1-1/2 LSB down from the desired analog full-scale voltage range and then adjusting the magnitude of the $V_{REF}/2$ input (pin 9) for a digital output code which is just changing from 1111 1110 to 1111 1111. When offsetting the zero and using a span adjusted $V_{REF}/2$ voltage, the full-scale adjustment is made by inputting V_{MIN} to the V_{IN} (−) input of the A/D and applying a voltage to the V_{IN} (+) input which is given by:

$$V_{IN}(+)\ \text{fs adj} = V_{MAX} - 1.5\left[\frac{(V_{MAX}-V_{MIN})}{256}\right]$$

where:

V_{MAX} = The high end of the analog input range

and

V_{MIN} = the low end (the offset zero) of the analog range. (Both are ground referenced.)

2.6 Clocking Option

The clock for the A/D can be derived from the CPU clock or an external RC can be added to provide self-clocking. The CLK IN (pin 4) makes use of a Schmitt trigger as shown in *Figure 4.*

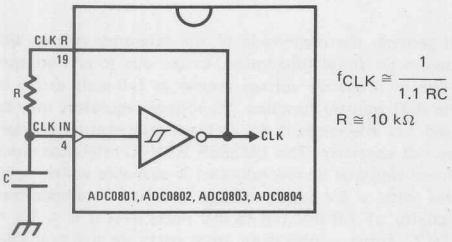


FIGURE 4. Self-Clocking the A/D

Heavy capacitive or DC loading of the clock R pin should be avoided as this will disturb normal converter operation. Loads less than 50 pF, such as driving up to 7 A/D converter clock inputs from a single clock R pin of 1 converter, are allowed. For larger clock line loading, a CMOS or low power T^2L buffer or PNP input logic should be used to minimize the loading on the clock R pin (do not use a standard T^2L buffer).

2.7 Restart During a Conversion

If the A/D is restarted ($\overline{CS}$ and $\overline{WR}$ go low and return high) during a conversion, the converter is reset and a new conversion is started. The output data latch is not updated if the conversion in process is not allowed to be completed, therefore the data of the previous conversion remains in this latch.

2.8 Continuous Conversions

For operation in the free-running mode an initializing pulse should be used, following power-up, to insure circuit operation. In this application, the $\overline{CS}$ input is grounded and the $\overline{WR}$ input is tied to the $\overline{INTR}$ output. This $\overline{WR}$ and $\overline{INTR}$ node should be momentarily forced to logic low following a power-up cycle to guarantee operation.

2.9 Driving the Data Bus

This MOS A/D, like MOS microprocessors and memories, will require a bus driver when the total capacitance of the data bus gets large. Other circuitry, which is tied to the data bus, will add to the total capacitive loading, even in TRI-STATE (high impedance mode). Backplane bussing also greatly adds to the stray capacitance of the data bus.

There are some alternatives available to the designer to handle this problem. Basically, the capacitive loading of the data bus slows down the response time, even though DC specifications are still met. For systems operating with a relatively slow CPU clock frequency, more time is available in which to establish proper logic levels on the bus and therefore higher capacitive loads can be driven (see typical characteristics curves).

At higher CPU clock frequencies time can be extended for I/O reads (and/or writes) by inserting wait states (8080) or using clock extending circuits (6800).

Finally, if time is short and capacitive loading is high, external bus drivers must be used. These can be TRI-STATE buffers (low power Schottky is recommended such as the DM74LS240 series) or special higher drive current products which are designed as bus drivers. High current bipolar bus drivers with PNP inputs are recommended.

2.10 Power Supplies

Noise spikes on the V_{CC} supply line can cause conversion errors as the comparator will respond to this noise. A low inductance tantalum filter capacitor should be used close to the converter V_{CC} pin and values of 1 μF or greater are recommended. If an unregulated voltage is available in the system, a separate LM340LAZ-5.0, TO-92, 5V voltage regulator for the converter (and other analog circuitry) will greatly reduce digital noise on the V_{CC} supply.

2.11 Wiring and Hook-Up Precautions

Standard digital wire wrap sockets are not satisfactory for breadboarding this A/D converter. Sockets on PC boards can be used and all logic signal wires and leads should be grouped and kept as far away as possible from the analog signal leads. Exposed leads to the analog inputs can cause undesired digital noise and hum pickup, therefore shielded leads may be necessary in many applications.

A single point analog ground should be used which is separate from the logic ground points. The power supply bypass capacitor and the self-clocking capacitor (if used) should both be returned to digital ground. Any $V_{REF}/2$ bypass capacitors, analog input filter capacitors, or input signal shielding should be returned to the analog ground point. A test for proper grounding is to measure the zero error of the A/D converter. Zero errors in excess of 1/4 LSB can usually be traced to improper board layout and wiring (see section 2.5.1 for measuring the zero error).

3.0 TESTING THE A/D CONVERTER

There are many degrees of complexity associated with testing an A/D converter. One of the simplest tests is to apply a known analog input voltage to the converter and use LEDs to display the resulting digital output code as shown in *Figure 5*.

For ease of testing, the $V_{REF}/2$ (pin 9) should be supplied with 2.560 V_{DC} and a V_{CC} supply voltage of 5.12 V_{DC} should be used. This provides an LSB value of 20 mV.

If a full-scale adjustment is to be made, an analog input voltage of 5.090 V_{DC} (5.120 − 1 1/2 LSB) should be applied to the $V_{IN}(+)$ pin with the $V_{IN}(-)$ pin grounded. The value of the $V_{REF}/2$ input voltage should then be adjusted until the digital output code is just changing from 1111 1110 to 1111 1111. This value of $V_{REF}/2$ should then be used for all the tests.

The digital output LED display can be decoded by dividing the 8 bits into 2 hex characters, the 4 most significant (MS) and the 4 least significant (LS). Table I shows the fractional binary equivalent of these two 4-bit groups. By adding the decoded voltages which are obtained from the column: Input voltage value for a 2.560 $V_{REF}/2$ of both the MS and the LS groups, the value of the digital display can be determined. For example, for an output LED display of 1011 0110 or B6 (in hex), the voltage values from the table are 3.520 + 0.120 or 3.640 V_{DC}. These voltage values represent the center-values of a perfect A/D converter. The effects of quantization error have to be accounted for in the interpretation of the test results.

For a higher speed test system, or to obtain plotted data, a digital-to-analog converter is needed for the test set-up. An accurate 10-bit DAC can serve as the precision voltage source for the A/D. Errors of the A/D under test can be provided as either analog voltages or differences in 2 digital words.

A basic A/D tester which uses a DAC and provides the error as an analog output voltage is shown in *Figure 6*. The 2 op amps can be eliminated if a lab DVM with a numerical subtraction feature is available to directly readout the difference voltage, "A–C". The analog input voltage can be supplied by a low frequency ramp generator and an X-Y plotter can be used to provide analog error (Y axis) versus analog input (X axis). The construction details of a tester of this type are provided in the NSC application note AN-179, "Analog-to-Digital Converter Testing".

For operation with a microprocessor or a computer-based test system, it is more convenient to present the errors digitally. This can be done with the circuit of *Figure 7*, where the output code transitions can be detected as the 10-bit DAC is incremented. This provides 1/4 LSB steps for the 8-bit A/D under test. If the results of this test are automatically plotted with the analog input on the X axis and the error (in LSB's) as the Y axis, a useful transfer function of the A/D under test results. For acceptance testing, the plot is not necessary and the testing speed can be increased by establishing internal limits on the allowed error for each code.

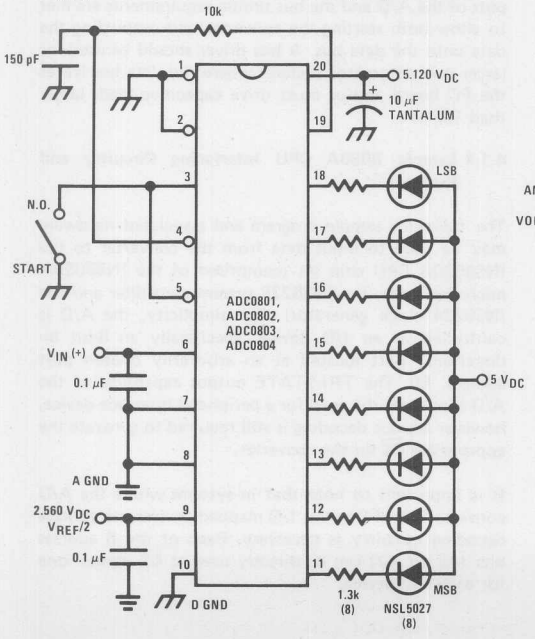


FIGURE 5. Basic A/D Tester

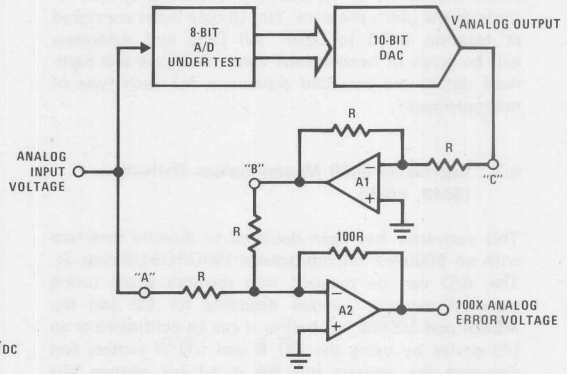


FIGURE 6. A/D Tester with Analog Error Output

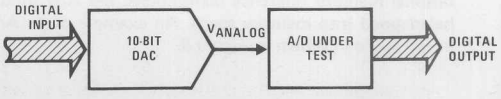


FIGURE 7. Basic "Digital" A/D Tester

TABLE I. DECODING THE DIGITAL OUTPUT LEDs

HEX	BINARY	FRACTIONAL BINARY VALUE FOR		OUTPUT VOLTAGE CENTER VALUES WITH $V_{REF}/2 = 2.560\ V_{DC}$	
		MS GROUP	LS GROUP	VMS GROUP*	VLS GROUP*
F	1 1 1 1	15/16	15/256	4.800	0.300
E	1 1 1 0	7/8	7/128	4.480	0.280
D	1 1 0 1	13/16	13/256	4.160	0.260
C	1 1 0 0	3/4	3/64	3.840	0.240
B	1 0 1 1	11/16	11/256	3.520	0.220
A	1 0 1 0	5/8	5/128	3.200	0.200
9	1 0 0 1	9/16	9/256	2.880	0.180
8	1 0 0 0	1/2	1/32	2.560	0.160
7	0 1 1 1	7/16	7/256	2.240	0.140
6	0 1 1 0	3/8	3/128	1.920	0.120
5	0 1 0 1	5/16	5/256	1.600	0.100
4	0 1 0 0	1/4	1/64	1.280	0.080
3	0 0 1 1	3/16	3/256	0.960	0.060
2	0 0 1 0	1/8	1/128	0.640	0.040
1	0 0 0 1	1/16	1/256	0.320	0.020
0	0 0 0 0			0	0

*V Display Output = VMS Group + VLS Group

4.0 MICROPROCESSOR INTERFACING

To discuss the interface with 8080A, 6800 and SC/MP-II microprocessors, a common sample subroutine structure is used. The microprocessor starts the A/D, reads and stores the results of 16 successive conversions, then returns to the user's program. The 16 data bytes are stored at location 0200 to 020F. All Data and Addresses will be given in hexadecimal form. Software and hardware details are provided separately for each type of microprocessor.

4.1 Interfacing 8080 Microprocessor Derivatives (8048, 8085)

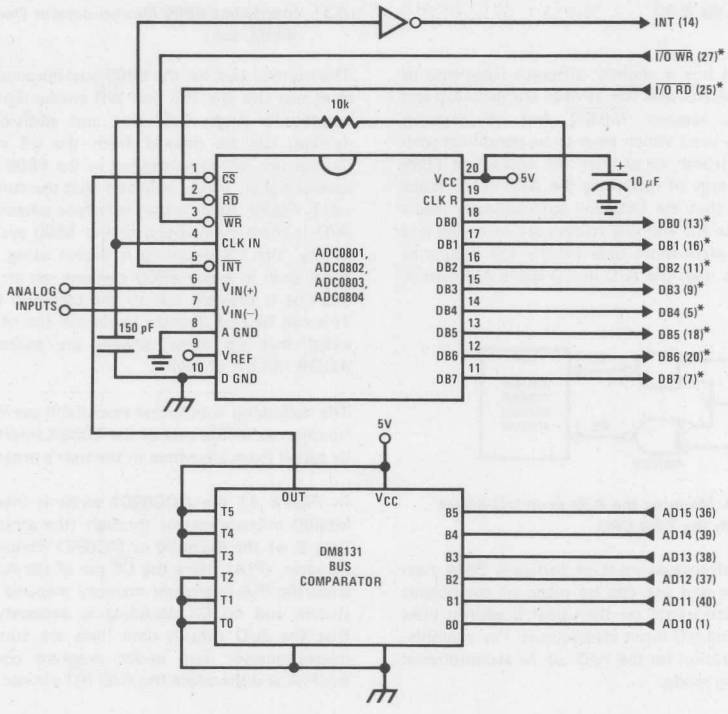
This converter has been designed to directly interface with an 8080A-2 microprocessor (MICROBUS class 2). The A/D can be mapped into memory space (using standard memory address decoding for $\overline{CS}$ and the $\overline{MEMR}$ and $\overline{MEMW}$ strobes) or it can be controlled as an I/O device by using the $\overline{I/O\ R}$ and $\overline{I/O\ W}$ strobes and decoding the address bits A0 → A7 (or address bits A8 → A15 as they will contain the same 8-bit address information) to obtain the $\overline{CS}$ input. Using the I/O space provides 256 additional addresses and may allow a simpler 8-bit address decoder but the data can only be input to the accumulator. To make use of the additional memory reference instructions, the A/D should be mapped into memory space. An example of an A/D in I/O space in shown in *Figure 8.*

The standard control bus signals of the 8080 ($\overline{CS}$, $\overline{RD}$ and $\overline{WR}$) can be directly wired to the digital control inputs of the A/D and the bus timing requirements are met to allow both starting the converter and outputting the data onto the data bus. A bus driver should be used for larger microprocessor systems where the data bus leaves the PC board and/or must drive capacitive loads larger than 100 pF.

4.1.1 Sample 8080A CPU Interfacing Circuitry and Program

The following sample program and associated hardware may be used to input data from the converter to the INS8080A CPU chip set (comprised of the INS8080A microprocessor, the INS8228 system controller and the INS8224 clock generator). For simplicity, the A/D is controlled as an I/O device, specifically an 8-bit bidirectional port located at an arbitrarily chosen port address, E0. The TRI-STATE output capability of the A/D eliminates the need for a peripheral interface device, however address decoding is still required to generate the appropriate $\overline{CS}$ for the converter.

It is important to note that in systems where the A/D converter is 1-of-8 or less I/O mapped devices, no address decoding circuitry is necessary. Each of the 8 address bits (A0 to A7) can be directly used as $\overline{CS}$ inputs—one for each I/O device.



Note 1: *Pin numbers for the INS8228 system controller, others are INS8080A.

Note 2: Pin 23 of the INS8228 must be tied to +12V through a 1 kΩ resistor to generate the RST 7 instruction when an interrupt is acknowledged as required by the accompanying sample program.

FIGURE 8. ADC0801–INS8080A CPU Interface

SAMPLE PROGRAM FOR *FIGURE 8* ADC0801–INS8080A CPU INTERFACE

```
0038    C3 00 03    RST 7:      JMP     LD DATA
  .        .          .
  .        .          .
0100    21 00 02    START:      LXI H 0200H     ; HL pair will point to
                                                ; data storage locations
0103    31 00 04    RETURN:     LXI SP 0400H    ; Initialize stack pointer (Note 1)
0106    7D                      MOV A, L        ; Test # of bytes entered
0107    FE OF                   CPI OF H        ; If # = 16.  JMP to
0109    CA 13 01                JZ CONT         ; user program
010C    D3 E0                   OUT E0 H        ; Start A/D
010E    FB                      EI              ; Enable interrupt
010F    00          LOOP:       NOP             ; Loop until end of
0110    C3 0F 01                JMP LOOP        ; conversion
0113       .        CONT:          .
  .        .          .              .
  .        .       (User program to  .
  .        .       process data)     .
  .        .          .              .
  .        .          .              .
0300    DB E0       LD DATA:    IN E0 H         ; Load data into accumulator
0302    77                      MOV M, A        ; Store data
0303    23                      INX H           ; Increment storage pointer
0304    C3 03 01                JMP  RETURN
```

Note 1: The stack pointer must be dimensioned because a RST 7 instruction pushes the PC onto the stack.

Note 2: All addresses used were arbitrarily chosen.

4.2 Interfacing the Z-80

The Z-80 control bus is slightly different from that of the 8080. General $\overline{RD}$ and $\overline{WR}$ strobes are provided and separate memory request, $\overline{MREQ}$, and I/O request, $\overline{IORQ}$, signals are used which have to be combined with the generalized strobes to provide the equivalent 8080 signals. An advantage of operating the A/D in I/O space with the Z-80 is that the CPU will automatically insert one wait state (the $\overline{RD}$ and $\overline{WR}$ strobes are extended one clock period) to allow more time for the I/O devices to respond. Logic to map the A/D in I/O space is shown in *Figure 9*.

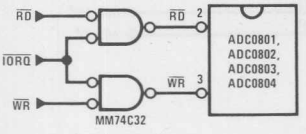


FIGURE 9. Mapping the A/D as an I/O device for use with the Z-80 CPU

Additional I/O advantages exist as software DMA routines are available and use can be made of the output data transfer which exists on the upper 8 address lines (A8 to A15) during I/O input instructions. For example, MUX channel selection for the A/D can be accomplished with this operating mode.

4.3 Interfacing 6800 Microprocessor Derivatives (6502, etc.)

The control bus for the 6800 microprocessor derivatives does not use the $\overline{RD}$ and $\overline{WR}$ strobe signals. Instead it employs a single R/$\overline{W}$ line and additional timing, if needed, can be derived from the $\phi 2$ clock. All I/O devices are memory mapped in the 6800 system, and a special signal, VMA, indicates that the current address is valid. *Figure 10* shows an interface schematic where the A/D is memory mapped in the 6800 system. For simplicity, the $\overline{CS}$ decoding is shown using 1/2 DM8092. Note that in many 6800 systems, an already decoded $\overline{4/5}$ line is brought out to the common bus at pin 21. This can be tied directly to the $\overline{CS}$ pin of the A/D, provided that no other devices are addressed at HEX ADDR: 4XXX or 5XXX.

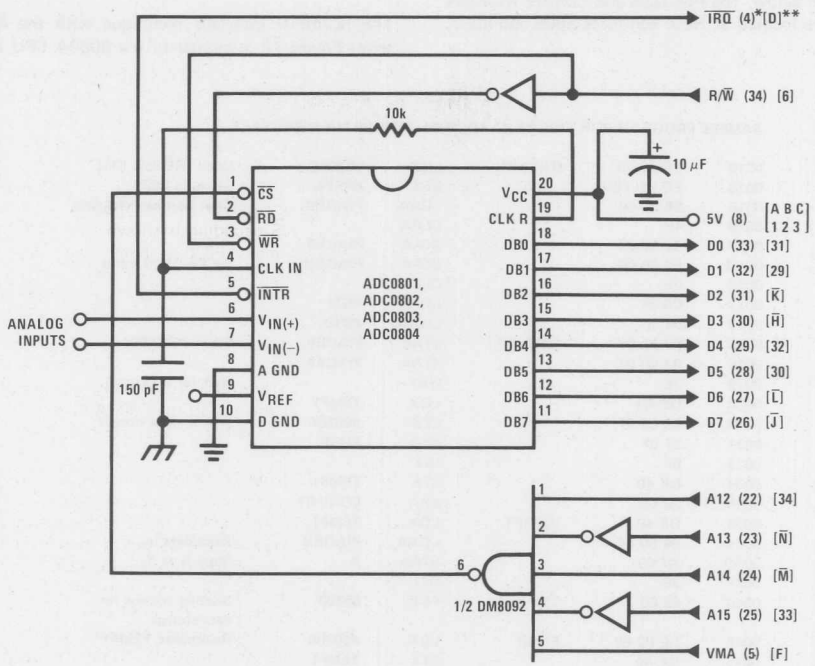
The following subroutine essentially performs the same function as in the case of the 8080A interface and it can be called from anywhere in the user's program.

In *Figure 11* the ADC0801 series is interfaced to the M6800 microprocessor through (the arbitrarily chosen) Port B of the MC6820 or MC6821 Peripheral Interface Adapter, (PIA). Here the $\overline{CS}$ pin of the A/D is grounded since the PIA is already memory mapped in the M6800 system and no $\overline{CS}$ decoding is necessary. Also notice that the A/D output data lines are connected to the microprocessor bus under program control through the PIA and therefore the A/D $\overline{RD}$ pin can be grounded.

SAMPLE PROGRAM FOR *FIGURE 10* ADC0801–MC6800 CPU INTERFACE

```
0010  DF 36     DATAIN  STX   TEMP2    ; Save contents of X
0012  CE 00 2C           LDX   #$002C   ; Upon IRQ low CPU
0015  FF FF F8           STX   $FFF8    ; jumps to 002C
0018  B7 50 00           STAA  $5000    ; Starts ADC0801
001B  0E                 CLI
001C  3E       CONVRT    WAI            ; Wait for interrupt
001D  DE 34              LDX   TEMP1
001F  8C 02 0F           CPX   #$020F   ; Is final data stored?
0022  27 14              BEQ   ENDP
0024  B7 50 00           STAA  $5000    ; Restarts ADC0801
0027  08                 INX
0028  DF 34              STX   TEMP1
002A  20 F0              BRA   CONVRT
002C  DE 34    INTRPT    LDX   TEMP1
002E  B6 50 00           LDAA  $5000    ; Read data
0031  A7 00              STAA  X        ; Store it at X
0033  3B                 RTI
0034  02 00    TEMP1     FDB   $0200    ; Starting address for
                                        ; data storage
0036  00 00    TEMP2     FDB   $0000
0038  CE 02 00 ENDP      LDX   #$0200   ; Reinitialize TEMP1
003B  DF 34              STX   TEMP1
003D  DE 36              LDX   TEMP2
003F  39                 RTS            ; Return from subroutine
                                        ; To user's program
```

Note 1: In order for the microprocessor to service subroutines and interrupts, the stack pointer must be dimensioned in the user's program.



Note 1: Numbers in parentheses refer to MC6800 CPU pin out.

Note 2: Numbers or letters in brackets refer to standard M6800 system common bus code.

FIGURE 10. ADC0801 – MC6800 CPU Interface

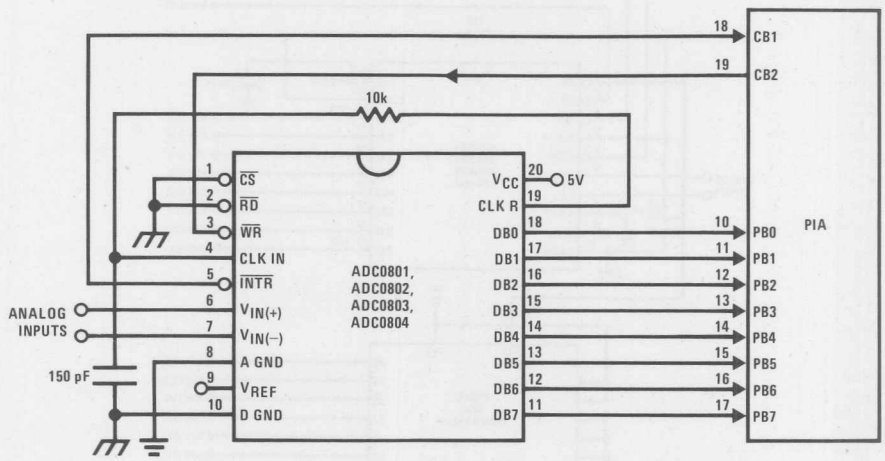


FIGURE 11. ADC0801–MC6820 PIA Interface

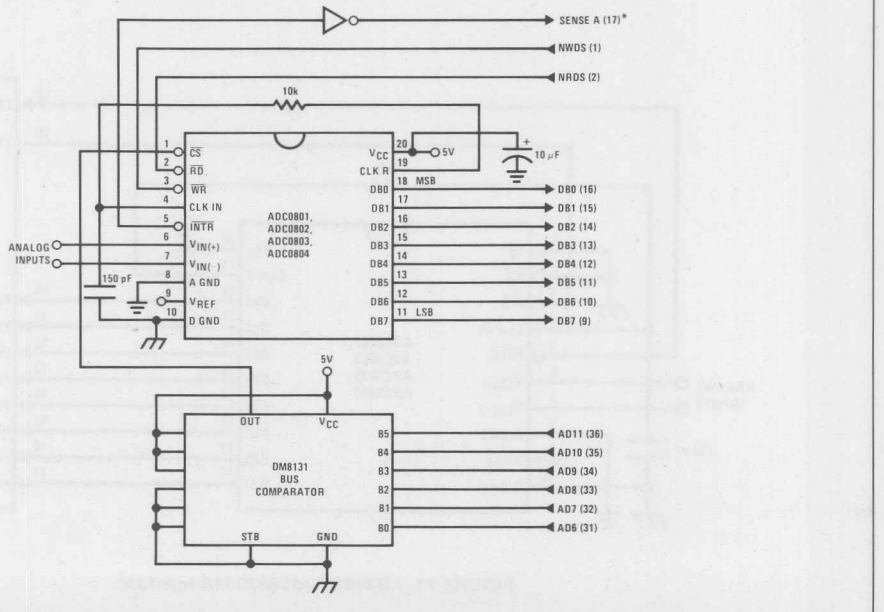
A sample interface program equivalent to the previous one, is shown below. The PIA Data and Control Registers of Port B are located at HEX addresses 8006 and 8007, respectively.

4.4 Interfacing the INS8060–SC/MP-II

The SC/MP-II interface technique with the ADC0801 series *Figure 12*, is similar to the 8080A CPU interface.

SAMPLE PROGRAM FOR *FIGURE 11* ADC0801–MC6820 PIA INTERFACE

```
0010  CE 00 38   DATAIN   LDX   #$0038   ; Upon IRQ low CPU
0013  FF FF F8            STX   $FFF8   ; jumps to 0038
0016  B6 80 06            LDAA  PIAORB  ; Clear possible IRQ flags
0019  4F                  CLRA
001A  B7 80 07            STAA  PIACRB
001D  B7 80 06            STAA  PIAORB  ; Set Port B as input
0020  0E                  CLI
0021  C6 34               LDAB  #$34
0023  86 3D               LDAA  #$3D
0025  F7 80 07   CONVRT   STAB  PIACRB  ; Starts ADC0801
0028  B7 80 07            STAA  PIACRB
002B  3E                  WAI           ; Wait for interrupt
002C  DE 40               LDX   TEMP1
002E  8C 02 0F            CPX   #$020F  ; Is final data stored?
0031  27 0F               BEQ   ENDP
0033  08                  INX
0034  DF 40               STX   TEMP1
0036  20 ED               BRA   CONVRT
0038  DE 40      INTRPT   LDX   TEMP1
003A  B6 80 06            LDAA  PIAORB  ; Read data in
003D  A7 00               STAA  X       ; Store it at X
003F  3B                  RTI
0040  02 00      TEMP1    FDB   $0200   ; Starting address for
                                        ; data storage
0042  CE 02 00   ENDP     LDX   #$0200  ; Reinitialize TEMP1
0045  DF 40               STX   TEMP1
0047  39                  RTS           ; Return from subroutine
                PIAORB    EQU   $8006   ; To user's program
                PIACRB    EQU   $8007
```



*Pin numbers in parentheses are for the SC/MP CPU.

FIGURE 12. ADC0801 – SC/MP-II Microprocessor Interface

The A/D is treated as a peripheral and it is mapped into the memory space of the SC/MP-II system. An address, 0D00, is assigned to the A/D and the $\overline{CS}$ signal is shown to be decoded by a bus comparator, DM8131. The $\overline{RD}$ and $\overline{WR}$ pins of the A/D are tied directly to the Write Data Strobe, NWRS, and Read Data Strobe, NRDS, pins of the SC/MP-II CPU. Notice that the $\overline{INTR}$ signal should be inverted before being tied to the SENSE A pin of the SC/MP-II. A sample interface program is shown below.

5.0 GENERAL APPLICATIONS

The following applications show some interesting uses for the A/D. The fact that one particular microprocessor is used is not meant to be restrictive. Each of these application circuits would have its counterpart using any microprocessor which is desired.

5.1 Multiple ADC0801 Series to MC6800 CPU Interface

To transfer analog data from several channels to a single microprocessor system, a multiple converter scheme presents several advantages over the conventional multiplexer single-converter approach. With the ADC0801 series, the differential inputs allow individual span adjustment for each channel. Furthermore, all analog input channels are sensed simultaneously, which essentially divides the microprocessor's total system servicing time by the number of channels, since all conversions occur simultaneously. This scheme is shown in *Figure 13*.

SAMPLE PROGRAM FOR *FIGURE 12* ADC0801–SC/MP-II MICROPROCESSOR INTERFACE

```
0100        08              NOP
0101    C4  02              LDI02
0103        35              XPAH(P1)
0104    C4  0D              LDI0D
0106        36              XPAH(P2)
0107    C4  03              LDI03
0109        37              XPAH(P3)
010A    C4  00              LDI00
010C        31              XPAL(P1)        ; P1=0200, P1 points to 1st byte address
010D    C4  00              LDI00
010F    C9  11              ST(P1+11)       ; Zero the byte count in address 0211
0112        32              XPAL(P2)        ; P2=0D00, P2 points to A/D
0113    CA  00  START:      ST(P2)          ; START the A/D
0115    C4  00              LDI00
0117        33              XPAL(P3)        ; P3=0300, P3 points to DATA in sub.
0118        05              IEN             ; starting address
0119        08  LOOP:       NOP
011A    90  FE              JMP(LOOP)
                User's Program
011C            USER        NOP
011D                        NOP
                              .
                              .
                              .
0300    C2  00  DATA IN:    LD(P2)          ; Load A/D data into accumulator
0302    CD  01              ST@1(P1)        ; Store A/D data and increment byte
                                            ; address
0304    A9  11              1LD(P1+11)      ; Increment byte count
0306    C4  0F              LDI0F
0308        03              SCL
0309    F9  11              CAD(P1+11)      ; 0F–(P1+11): Is byte count = 16?
030B    9B  03              JZ(USER)        ; If byte count = 16 jump to user's
                                            ; program
030D    C4  13              LDI13
030F        33              XPAL(P3)        ; P3=0113
0310        3F              XPPC(P3)        ; Go to START and do another conversion
```

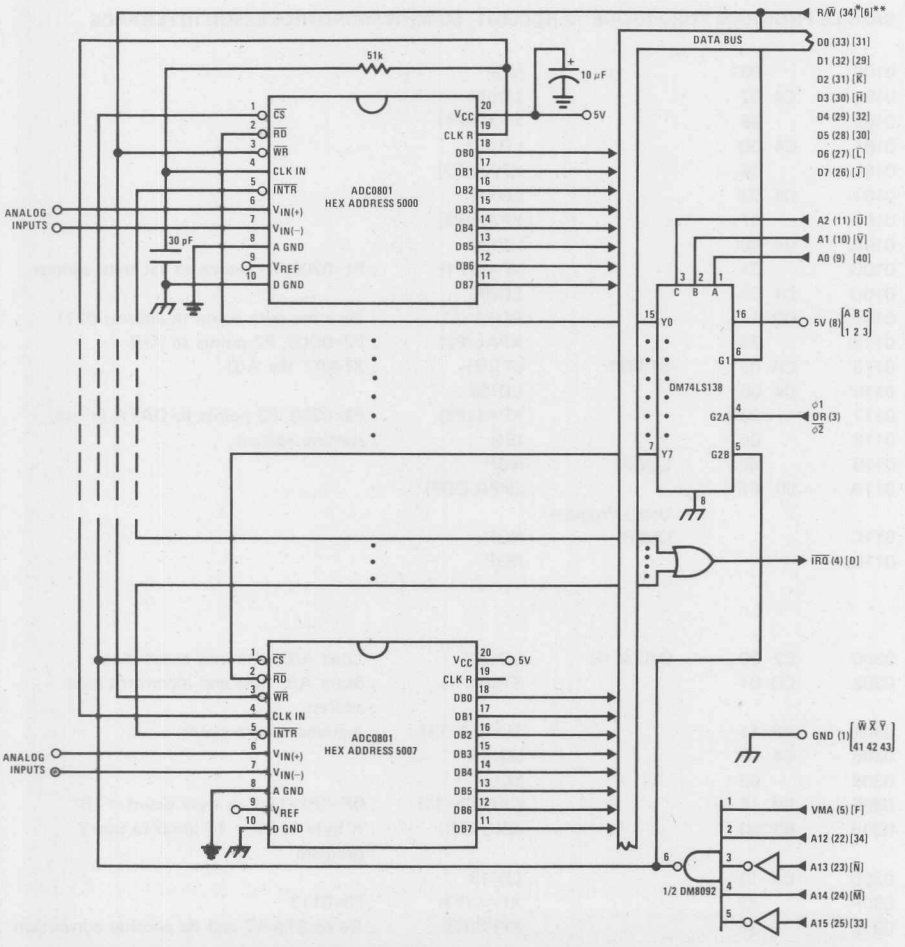
The following schematic and sample subroutine (DATA IN) may be used to interface (up to) 8 ADC0801's directly to the MC6800 CPU. This scheme can easily be extended to allow the interface of more converters. In this configuration the converters are (arbitrarily) located at HEX address 5000 in the MC6800 memory space. To save components, the clock signal is derived from just one RC pair on the first converter. This output drives the other A/Ds.

All the converters are started simultaneously with a STORE instruction at HEX address 5000. Note that any other HEX address of the form 5XXX will be decoded by the circuit, pulling all the $\overline{CS}$ inputs low. This can easily be avoided by using a more definitive address decoding scheme. All the interrupts are ORed together to insure that all A/Ds have completed their conversion before the microprocessor is interrupted.

The subroutine, DATA IN, may be called from anywhere in the user's program. Once called, this routine initializes the CPU, starts all the converters simultaneously and waits for the interrupt signal. Upon receiving the interrupt, it reads the converters (from HEX addresses 5000 through 5007) and stores the data successively at (arbitrarily chosen) HEX addresses 0200 to 0207, before returning to the user's program. All CPU registers then recover the original data they had before servicing DATA IN.

5.2 Auto-Zeroed Differential Transducer Amplifier and A/D Converter

The differential inputs of the ADC0801 series eliminate the need to perform a differential to single ended conversion for a differential transducer. Thus, one op amp can be eliminated since the differential to single ended conversion is provided by the differential input of the ADC0801 series. In general, a transducer preamp is required to take advantage of the full A/D converter input dynamic range.



Note 1: Numbers in parentheses refer to MC6800 CPU pin out.

Note 2: Numbers or letters in brackets refer to standard M6800 system common bus code.

FIGURE 13. Interfacing Multiple A/Ds in a MC6800 System

PROGRAM FOR *FIGURE 13* INTERFACING MULTIPLE A/Ds IN AN MC6800 SYSTEM

ADDRESS	HEX CODE	MNEMONICS			COMMENTS
0010	DF 44	DATAIN	STX	TEMP	;Save Contents of X
0012	CE 00 2A		LDX	#$002A	;Upon $\overline{IRQ}$ LOW CPU
0015	FF FF F8		STX	$FFF8	;Jumps to 002A
0018	B7 50 00		STAA	$5000	;Starts all A/D's
001B	0E		CLI		
001C	3E		WAI		;Wait for interrupt
001D	CE 50 00		LDX	#$5000	
0020	DF 40		STX	INDEX1	;Reset both INDEX
0022	CE 02 00		LDX	#$0200	;1 and 2 to starting
0025	DF 42		STX	INDEX2	;addresses
0027	DE 44		LDX	TEMP	
0029	39		RTS		;Return from subroutine
002A	DE 40	INTRPT	LDX	INDEX1	;INDEX1 → X
002C	A6 00		LDAA	X	;Read data in from A/D at X
002E	08		INX		;Increment X by one
002F	DF 40		STX	INDEX1	;X → INDEX1
0031	DE 42		LDX	INDEX2	;INDEX2 → X
0033	A7 00		STAA	X	;Store data at X
0035	8C 02 07		CPX	#$0207	;Have all A/D's been read?
0038	27 05		BEQ	RETURN	;Yes: branch to RETURN
003A	08		INX		;No: increment X by one
003B	DF 42		STX	INDEX2	;X → INDEX2
003D	20 EB		BRA	INTRPT	;Branch to 002A
003F	3B	RETURN	RTI		
0040	50 00	INDEX1	FDB	$5000	;Starting address for A/D
0042	02 00	INDEX2	FDB	$0200	;Starting address for data storage
0044	00 00	TEMP	FDB	$0000	

Note 1: In order for the microprocessor to service subroutines and interrupts, the stack pointer must be dimensioned in the user's program.

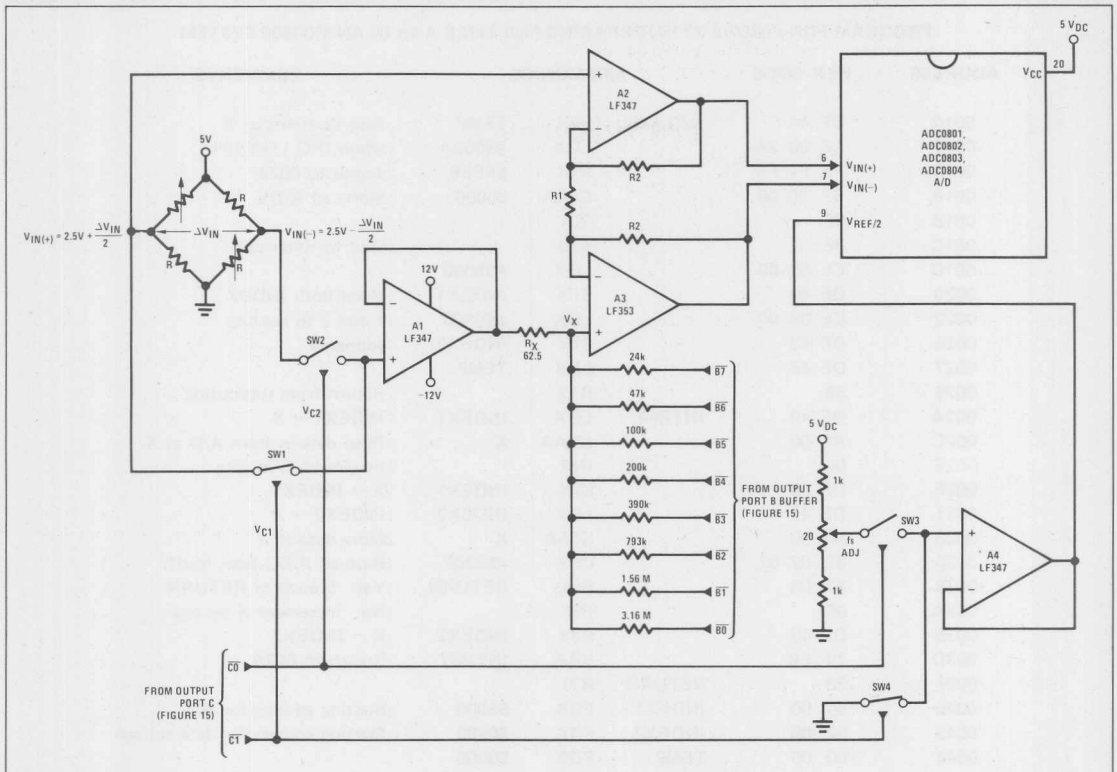
For amplification of DC input signals, a major system error is the input offset voltage of the amplifiers used for the preamp. *Figure 14* is a gain of 100 differential preamp whose offset voltage errors will be cancelled by a zeroing subroutine which is performed by the INS8080A microprocessor system. The total allowable input offset voltage error for this preamp is only 50 μV for 1/4 LSB error. This would obviously require very precise amplifiers. The expression for the differential output voltage of the preamp is:

$$V_o = \underbrace{[V_{IN}(+) - V_{IN}(-)]}_{\text{SIGNAL}} \underbrace{\left[1 + \frac{2R2}{R1}\right]}_{\text{GAIN}} + \underbrace{(V_{os2} - V_{os1} - V_{os3} \pm I_x R_x)}_{\text{DC ERROR TERM}} \underbrace{\left(1 + \frac{2R2}{R1}\right)}_{\text{GAIN}}$$

where I_x is the current through resistor R_x. All of the offset error terms can be cancelled by making $\pm I_x R_x = V_{os1} + V_{os3} - V_{os2}$. This is the principle of this auto-zeroing scheme.

The INS8080A uses the 3 I/O ports of an INS8255 Programmable Peripheral Interface (PPI) to control the auto zeroing and input data from the ADC0801 as shown in *Figure 15*. The PPI is programmed for basic I/O operation (mode 0) with Port A being an input port and Ports B and C being output ports. Two bits of Port C are used to alternately open or close the 2 switches at the input of the preamp. Switch SW1 is closed to force the preamp's differential input to be zero during the zeroing subroutine and then opened and SW2 is then closed for conversion of the actual differential input signal. Using 2 switches in this manner eliminates concern for the ON resistance of the switches as they must conduct only the input bias current of the input amplifiers.

Output Port B is used as a successive approximation register by the 8080 and the binary scaled resistors in series with each output bit create a D/A converter. During the zeroing subroutine, the voltage at V_X increases or decreases as required to make the differential output voltage equal to zero. This is accomplished by insuring that the voltage at the output of A1 is approximately 2.5V so that a logic "1" (5V) on any output of Port B will source current into node V_X thus raising the voltage at V_X and making the output differential more negative. Conversely, a logic "0" (0V) will pull current out of node V_X and decrease the voltage, causing the differential output to become more positive. For the resistor values shown, V_X can move ±12 mV with a resolution of 50 μV which will null the offset error term to 1/4 LSB of full-scale for the ADC0801. It is important that the voltage levels which drive the auto-zero resistors be constant. Also, for symmetry, a logic swing of 0V to 5V is convenient. To achieve this, a CMOS buffer is used for the logic output signals of Port B and this CMOS package is powered with a stable 5V source. Buffer amplifier A1 is necessary so that it can source or sink the D/A output current.



Note 1: R2 = 49.5 R1

Note 2: Switches are CD4066BC CMOS analog switches.

Note 3: The 9 resistors used in the auto-zero section can be ±5% tolerance.

FIGURE 14. Gain of 100 Differential Transducer Preamp

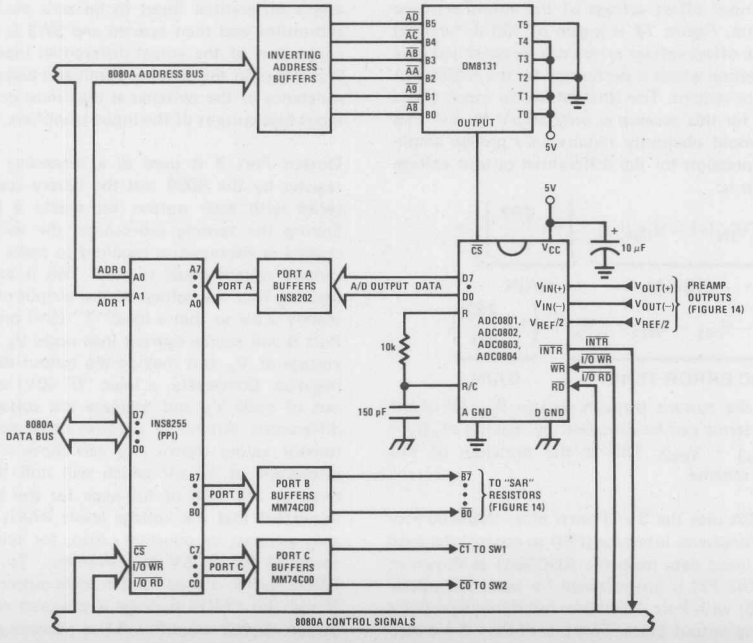


FIGURE 15. Microprocessor Interface Circuitry for Differential Preamp

A flow chart for the zeroing subroutine is shown in *Figure 16*. It must be noted that the ADC0801 series will output an all zero code when it converts a negative input [$V_{IN}(-) \geq V_{IN}(+)$]. Also, a logic inversion exists as all of the I/O ports are buffered with inverting gates.

Basically, if the data read is zero, the differential output voltage is negative, so a bit in Port B is cleared to pull V_X more negative which will make the output more positive for the next conversion. If the data read is not zero, the output voltage is positive so a bit in Port B is set to make V_X more positive and the output more negative. This continues for 8 approximations and the differential output eventually converges to within 5 mV of zero.

The actual program is given in *Figure 17*. All addresses used are compatible with the BLC 80/10 microcomputer system. In particular:

Port A and the ADC0801 are at port address E4
Port B is at port address E5
Port C is at port address E6
PPI control word port is at port address E7
Program Counter automatically goes to ADDR:3C3D upon acknowledgement of an interrupt from the ADC0801

5-3 Multiple A/D Converters in a Z-80 Interrupt Driven Mode

In data acquisition systems where more than one A/D converter (or other peripheral device) will be interrupting program execution of a microprocessor, there is obviously a need for the CPU to determine which device requires servicing. *Figure 18* and the accompanying software is a method of determining which of 7 ADC0801 converters has completed a conversion ($\overline{INTR}$ asserted) and is requesting an interrupt. This circuit allows starting the A/D converters in any sequence, but will input and store valid data from the converters with a priority sequence of A/D 1 being read first, A/D 2 second, etc., through A/D 7 which would have the lowest priority for data being read. Only the converters whose INT is asserted will be read.

The key to decoding circuitry is the MM74LS373, 8-bit D type flip-flop. When the Z-80 acknowledges the interrupt, the program is vectored to a data input Z-80 subroutine. This subroutine will read a peripheral status word from the MM74LS373 which contains the logic state of the $\overline{INTR}$ outputs of all the converters. Each converter which initiates an interrupt will place a logic "0" in a unique bit position in the status word and the subroutine will determine the identity of the converter and execute a data read. An identifier word (which indicates which A/D the data came from) is stored in the next sequential memory location above the location of the data so the program can keep track of the identity of the data entered.

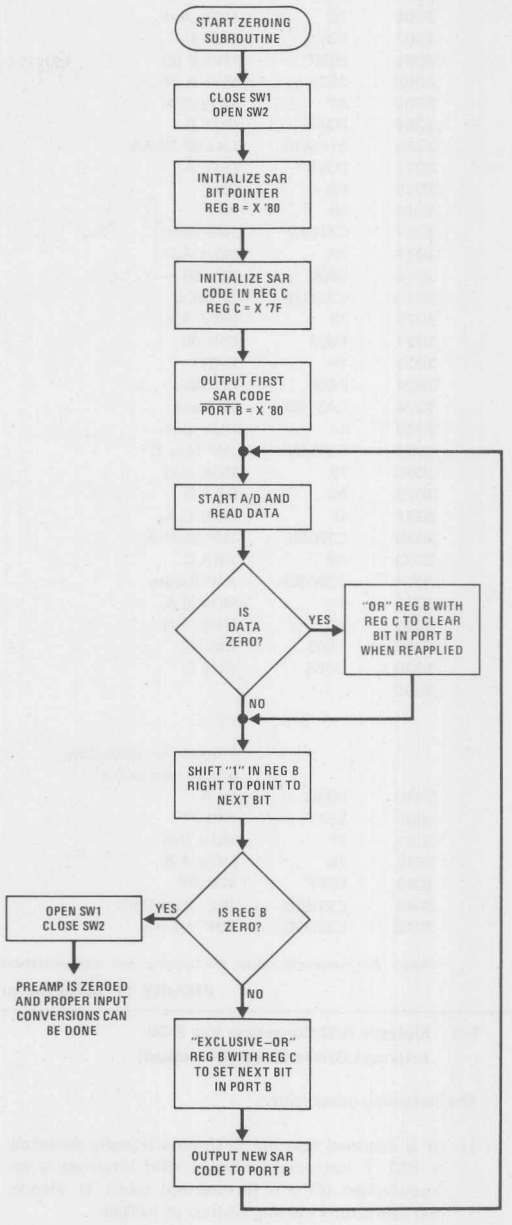


FIGURE 16. Flow Chart for Auto-Zero Routine

```
3D00    3E90    MVI 90
3D02    D3E7    Out Control Port                ; Program PPI
3D04    2601    MVI H 01        Auto-Zero Subroutine
3D06    7C      MOV A,H
3D07    D3E6    OUT C                           ; Close SW1, open SW2
3D09    0680    MVI B 80                        ; Initialize SAR bit pointer
3D0B    3E7F    MVI A 7F                        ; Initialize SAR code
3D0D    4F      MOV C,A         Return
3D0E    D3E5    OUT B                           ; Port B = SAR code
3D10    31AA3D  LXI SP 3DAA     Start           ; Dimension stack pointer
3D13    D3E4    OUT A                           ; Start A/D
3D15    FB      IE
3D16    00      NOP             Loop            ; Loop until INT asserted
3D17    C3163D  JMP Loop
3D1A    7A      MOV A,D         Auto-Zero
3D1B    C600    ADI 00
3D1D    CA2D3D  JZ Set C                        ; Test A/D output data for zero
3D20    78      MOV A,B         Shift B
3D21    F600    ORI 00                          ; Clear carry
3D23    1F      RAR                             ; Shift "1" in B right one place
3D24    FE00    CPI 00                          ; Is B zero? If yes last
3D26    CA373D  JZ Done                         ; approximation has been made
3D29    47      MOV B,A
3D2A    C3333D  JMP New C
3D2D    79      MOV A,C         Set C
3D2E    B0      ORA B                           ; Set bit in C that is in same
3D2F    4F      MOV C,A                         ; position as "1" in B
3D30    C3203D  JMP Shift B
3D33    A9      XRA C           New C           ; Clear bit in C that is in
3D34    C30D3D  JMP Return                      ; same position as "1" in B
3D37    47      MOV B,A         Done            ; then output new SAR code.
3D38    7C      MOV A,H                         ; Open SW1, close SW2 then
3D39    EE03    XRI 03                          ; proceed with program. Preamp
3D3B    D3E6    OUT C                           ; is now zeroed.
3D3D            .               Normal
                .
                .
                Program for processing
                proper data values
3C3D    DBE4    IN A            Read A/D Subroutine  ; Read A/D data
3C3F    EEFF    XRI FF                          ; Invert data
3C41    57      MOV D,A
3C42    78      MOV A,B                         ; Is B Reg = 0? If not stay
3C43    E6FF    ANI FF                          ; in auto zero subroutine
3C45    C21A3D  JNZ  Auto-Zero
3C48    C33D3D  JMP  Normal
```

Note: All numerical values are hexadecimal representations.

FIGURE 17. Software for Auto-Zeroed Differential A/D

5-3 Multiple A/D Converters in a Z-80 Interrupt Driven Mode (Continued)

The following notes apply:

1) It is assumed that the CPU automatically performs a RST 7 instruction when a valid interrupt is acknowledged (CPU is in interrupt mode 1). Hence, the subroutine starting address of X0038.

2) The address bus from the Z-80 and the data bus to the Z-80 are assumed to be inverted by bus drivers.

3) A/D data and identifying words will be stored in sequential memory locations starting at the arbitrarily chosen address X 3E00.

4) The stack pointer must be dimensioned in the main program as the RST 7 instruction automatically pushes the PC onto the stack and the subroutine uses an additional 6 stack addresses.

5) The peripherals of concern are mapped into I/O space with the following port assignments:

HEX PORT ADDRESS	PERIPHERAL
00	MM74C374 8-bit flip-flop
01	A/D 1
02	A/D 2
03	A/D 3
04	A/D 4
05	A/D 5
06	A/D 6
07	A/D 7

This port address also serves as the A/D identifying word in the program.

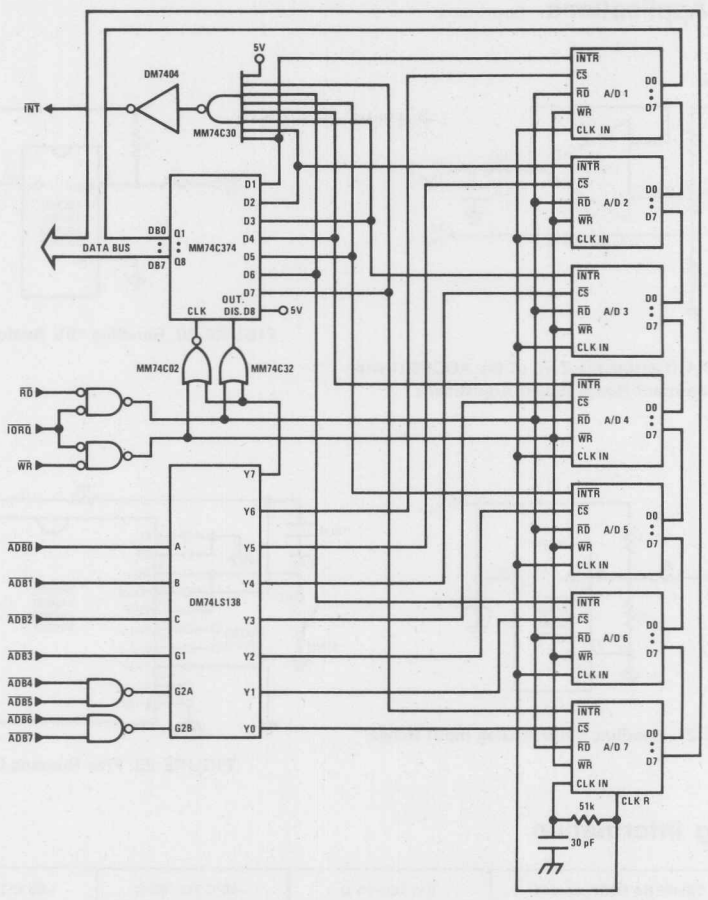


FIGURE 18. Multiple A/D's with Z-80 Type Microprocessor

INTERRUPT SERVICING SUBROUTINE

LOC	OBJ CODE		SOURCE STATEMENT	COMMENT
0038	E5		PUSH HL	; Save contents of all registers affected by
0039	C5		PUSH BC	; this subroutine.
003A	F5		PUSH AF	; Assumed INT mode 1 earlier set.
003B	21 00 3E		LD (HL),X3E00	; Initialize memory pointer where data will be stored.
003E	0E 01		LD C,X01	; C register will be port ADDR of A/D converters.
0040	D300		OUT X00,A	; Load peripheral status word into 8-bit latch.
0042	DB00		IN A, X00	; Load status word into accumulator.
0044	47		LD B,A	; Save the status word.
0045	79	TEST	LD A,C	; Test to see if the status of all A/D's have
0046	FE 08		CP, X08	; been checked. If so, exit subroutine.
0048	CA 60 00		JPZ, DONE	
004B	78		LD A,B	; Test a single bit in status word by looking for
004C	1F		RRA	; a "1" to be rotated into the CARRY (an $\overline{INT}$
004D	47		LD B,A	; is loaded as a "1"). If CARRY is set then load
004E	DA 5500		JPC, LOAD	; contents of A/D at port ADDR in C register.
0051	0C	NEXT	INC C	; If CARRY is not set, increment C register to point
0052	C3 4500		JP,TEST	; to next A/D, then test next bit in status word.
0055	ED 78	LOAD	IN A, (C)	; Read data from interrupting A/D and invert
0057	EE FF		XOR FF	; the data.
0059	77		LD (HL),A	; Store the data.
005A	2C		INC L	
005B	71		LD (HL),C	; Store A/D identifier (A/D port ADDR).
005C	2C		INC L	
005D	C3 51 00		JP,NEXT	; Test next bit in status word.
0060	F1	DONE	POP AF	; Re-establish all registers as they were
0061	C1		POP BC	; before the interrupt.
0062	E1		POP HL	
0063	C9		RET	; Return to original program.

Typical Applications (Continued)

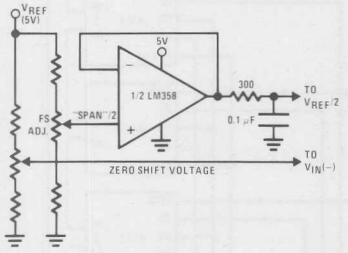


FIGURE 19. Offsetting the Zero of the ADC0801 and Performing an Input Range (Span) Adjustment

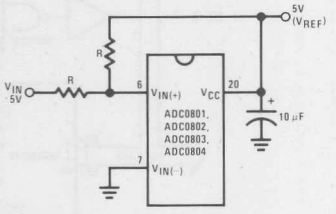


FIGURE 20. Handling ±5V Analog Input Range

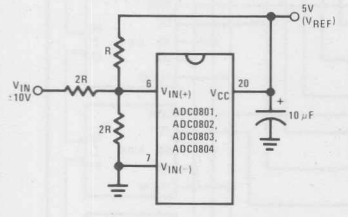


FIGURE 21. Handling ±10V Analog Input Range

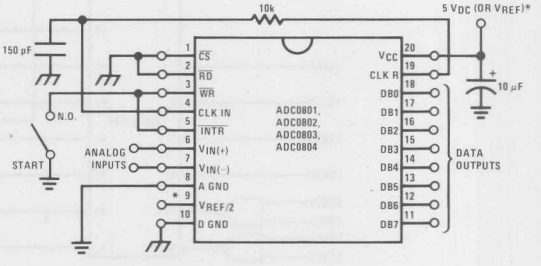


FIGURE 22. Free Running Connection

Ordering Information

TEMPERATURE RANGE		0 C TO +70 C	−40 C TO +85 C	−55 C TO +125 C
ERROR	±1/4 Bit Adjusted	ADC0801LCN	ADC0801LCD	ADC0801LD
	±1/2 Bit Unadjusted	ADC0802LCN	ADC0802LCD	ADC0802LD
	±1/2 Bit Adjusted	ADC0803LCN	ADC0803LCD	ADC0803LD
	±1 Bit Unadjusted	ADC0804LCN	ADC0804LCD	
PACKAGE OUTLINE		N20A–MOLDED DIP	D20A–CAVITY DIP	D20A–CAVITY DIP

National Semiconductor

ADC3511 3½-Digit Microprocessor Compatible A/D Converter
ADC3711 3¾-Digit Microprocessor Compatible A/D Converter

General Description

The ADC3511 and ADC3711 (MM74C937-1, MM74C938-1) monolithic A/D converter circuits are manufactured using standard complementary MOS (CMOS) technology. A pulse modulation analog-to-digital conversion technique is used and requires no external precision components. In addition, this technique allows the use of a reference voltage that is the same polarity as the input voltage.

One 5V (TTL) power supply is required. Operating with an isolated supply allows the conversion of positive as well as negative voltages. The sign of the input voltage is automatically determined and indicated on the sign pin. If the power supply is not isolated, only one polarity of voltage may be converted.

The conversion rate is set by an internal oscillator. The frequency of the oscillator can be set by an external RC network or the oscillator can be driven from an external frequency source. When using the external RC network, a square wave output is available.

The ADC3511 and ADC3711 have been designed to provide addressed BCD data and are intended for use with microprocessors and other digital systems. BCD digits are selected on demand via 2 Digit Select (D0, D1) inputs. Digit Select inputs are latched by a low-to-high transition on the Digit Latch Enable (DLE) input and will remain latched as long as DLE remains high. A start conversion input and a conversion complete output are included on both the ADC3511 and the ADC3711.

Features

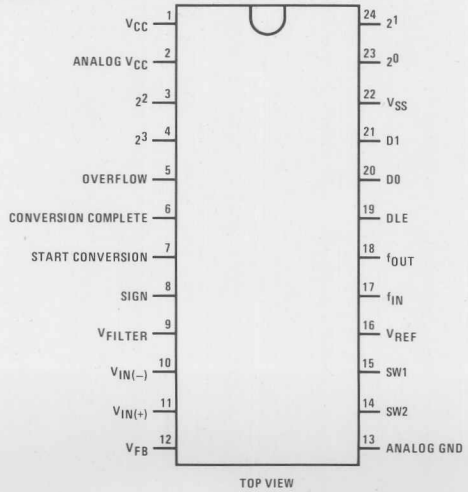
- Operates from single 5V supply
- ADC3511 converts 0 to ±1999 counts
- ADC3711 converts 0 to ±3999 counts
- Addressed BCD outputs
- No external precision components necessary
- Easily interfaced to microprocessors or other digital systems
- Medium speed—200 ms/conversion
- TTL compatible
- Internal clock set with RC network or driven externally
- Overflow indicated by hex "EEEE" output reading as well as an overflow output

Applications

- Low cost analog-to-digital converter
- Eliminate analog multiplexing by using remote A/D converters
- Convert analog transducers (temperature, pressure, displacement, etc.) to digital transducers

Connection Diagram

Dual-In-Line Package



TOP VIEW

Communications Components

National Semiconductor

October 1980

INS2651 Programmable Communications Interface

General Description

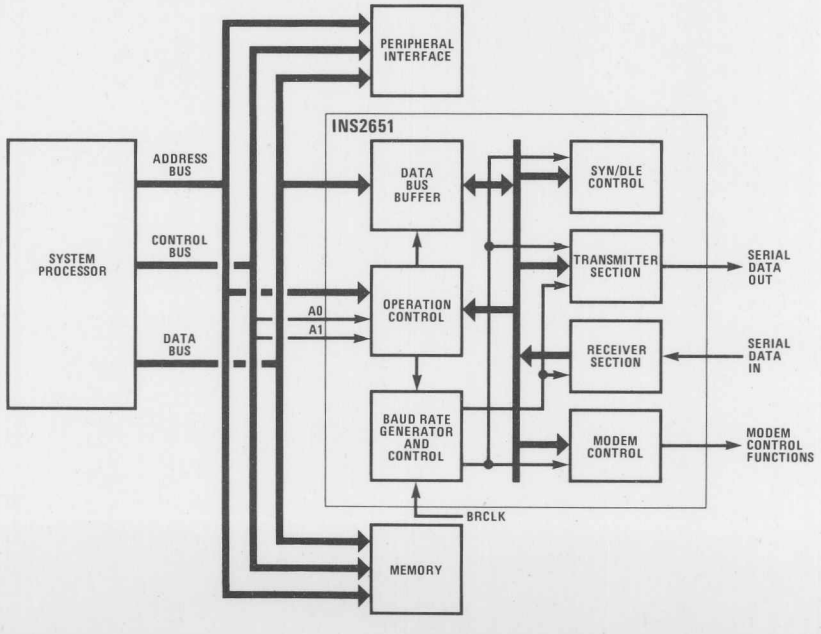
The INS2651 is a programmable Universal Synchronous/Asynchronous Receiver/Transmitter (USART) chip contained in a standard 28-pin dual-in-line package. The chip, which is fabricated using N-channel silicon gate MOS technology, functions as a serial data input/output interface in a bus structured system. The functional configuration of INS2651 is programmed by the system software for maximum flexibility, thereby allowing the system to receive and transmit virtually any serial data communications signal presently in use.

The INS2651 can be programmed to receive and transmit either synchronous or asynchronous serial data. The INS2651 performs serial-to-parallel conversion on data characters received from an input/output device or a MODEM, and parallel-to-serial conversion on data characters received from the CPU. The CPU can read the complete status of the INS2651 at any time during the functional operation. Status information reported includes the type and the condition of the transfer operations being performed by the INS2651, as well as error conditions (parity, overrun, or framing).

Features

- Synchronous and Asynchronous Full Duplex or Half Duplex Operations
- Synchronous Mode Capabilities
  - Selectable 5- to 8-Bit Characters
  - Selectable 1 or 2 SYNC Characters
  - Transparent or Non-Transparent Mode
  - Automatic SYNC or DLE-SYNC Insertion
  - SYNC or DLE Stripping
- Asynchronous Mode Capabilities
  - Selectable 5- to 8-Bit Characters
  - 3 Selectable Clock Rates (1x, 16x, or 64x the Baud Rate)
  - Line Break Detection and Generation
  - 1-, 1½-, or 2-Stop Bit Detection and Generation
  - False Start Bit Detection
- Baud Rates
  - DC to 0.8 M Baud (Synchronous)
  - DC to 0.8 M Baud (1x, Asynchronous)
  - DC to 50 k Baud (16x, Asynchronous)
  - DC to 12.5 k Baud (64x, Asynchronous)
- Internal or External Baud Rate Clock
  - 16 Internal Rates (50 to 19,200 Baud)
- Double Buffering of Data
- TTL Compatible
- No System Clock Required
- Direct Plug-In Replacement for Signetics 2651

INS2651 General System Configuration



Absolute Maximum Ratings

Operating Ambient Temperature	0°C to +70°C
Storage Temperature	−65°C to +150°C
All Voltages with Respect to Ground	−0.5 V to +6.0 V

Note: *Maximum ratings indicate limits beyond which permanent damage may occur. Continuous operation at these limits is not intended and should be limited to those conditions specified under DC Electrical Characteristics.*

DC Electrical Characteristics

T_A = 0°C to +70°C; V_{CC} = +5.0 V ± 5%, GND = 0 V

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage			0.8	V	
V_{IH}	Input High Voltage	2.0			V	
V_{OL}	Output Low Voltage		0.25	0.45	V	I_{OL} = 1.6 mA
V_{OH}	Output High Voltage	2.4	2.8		V	I_{OH} = −100 μA
I_{IL}	Input Load Current			10	μA	V_{IN} = 0 V to 5.5 V
I_{LD}	Data Bus Leakage Current			10	μA	V_{OUT} = 4.0 V
I_{LO}	Open Drain Leakage Current			10	μA	V_{OUT} = 4.0 V
I_{CC}	Power Supply Current		65	150	mA	

Capacitance

T_A = +25°C; V_{CC} = GND = 0 V

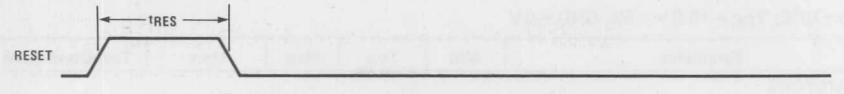
Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
C_{IN}	Input Capacitance			20	pF	f_c = 1 MHz
C_{OUT}	Output Capacitance			20	pF	Unmeasured pins to ground
$C_{I/O}$	I/O Capacitance			20	pF	

AC Electrical Characteristics

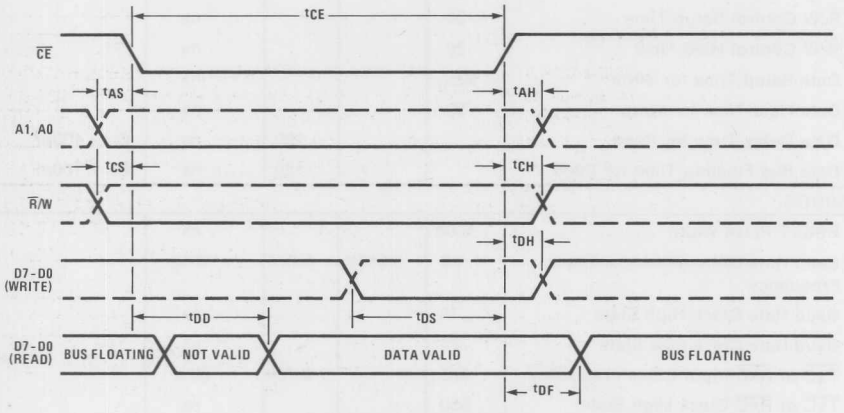
T_A = 0°C to +70°C; V_{CC} = +5.0 V ± 5%, GND = 0 V

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
BUS PARAMETERS						
t_{CE}	Chip Enable Pulse Width	300			ns	
t_{AS}	Address Setup Time	20			ns	
t_{AH}	Address Hold Time	20			ns	
t_{CS}	$\overline{R}$/W Control Setup Time	20			ns	
t_{CH}	$\overline{R}$/W Control Hold Time	20			ns	
t_{DS}	Data Setup Time for Write	225			ns	
t_{DH}	Data Hold Time for Write	50			ns	
t_{DD}	Data Delay Time for Read			250	ns	C_L = 100pF
t_{DF}	Data Bus Floating Time for Read			150	ns	C_L = 100pF
OTHER TIMINGS						
t_{RES}	RESET Pulse Width	1000			ns	
f_{BRG}	Baud Rate Generator Input Clock Frequency	1.0	5.0688	5.073	MHz	
t_{BRH}	Baud Rate Clock High State	70			ns	
t_{BRL}	Baud Rate Clock Low State	70			ns	
$f_{R/T}$	$\overline{TxC}$ or $\overline{RxC}$ Input Clock Frequency	DC		0.769	MHz	
$t_{R/TH}$	$\overline{TxC}$ or $\overline{RxC}$ Clock High State	650			ns	
$t_{R/TL}$	$\overline{TxC}$ or $\overline{RxC}$ Clock Low State	650			ns	
t_{TxD}	TxD Delay from Falling Edge of $\overline{TxC}$			650	ns	C_L = 100pF
t_{TCS}	Skew Between TxD Changing and Falling Edge of $\overline{TxC}$ Output		0	0	ns	C_L = 100pF
t_{RxS}	Rx Data Setup Time	300			ns	
t_{RxH}	Rx Data Hold Time	300			ns	

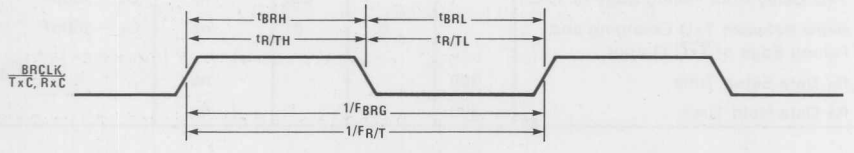
Timing Waveforms



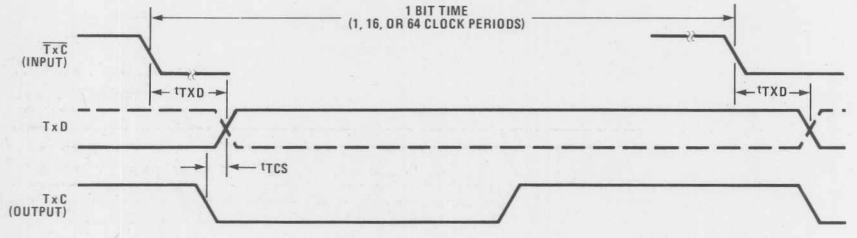
RESET TIMING



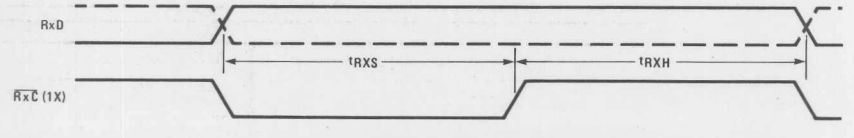
READ AND WRITE TIMING



CLOCK TIMING

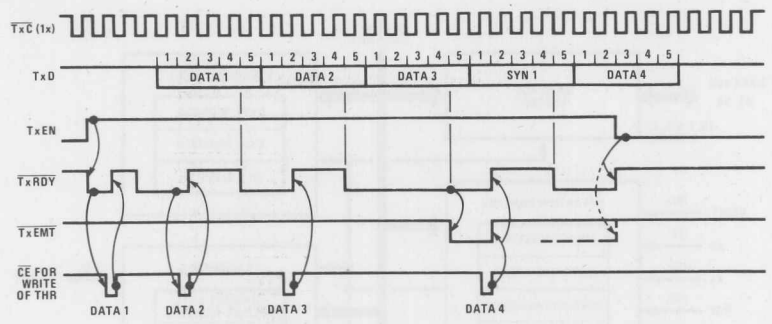


TRANSMIT TIMING

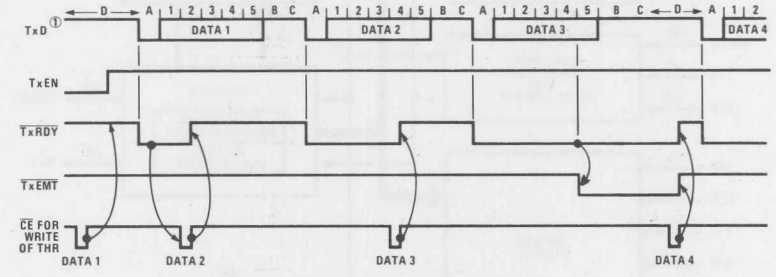


RECEIVE TIMING

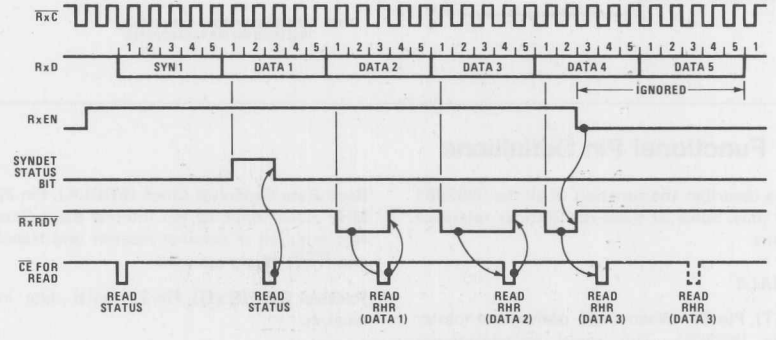
Timing Waveforms (cont'd.)



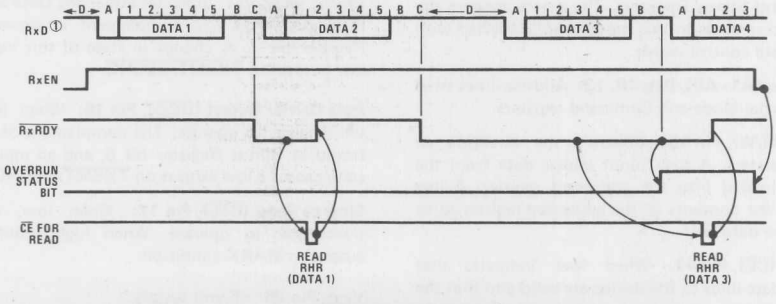
SYNCHRONOUS MODE



ASYNCHRONOUS MODE



SYNCHRONOUS MODE

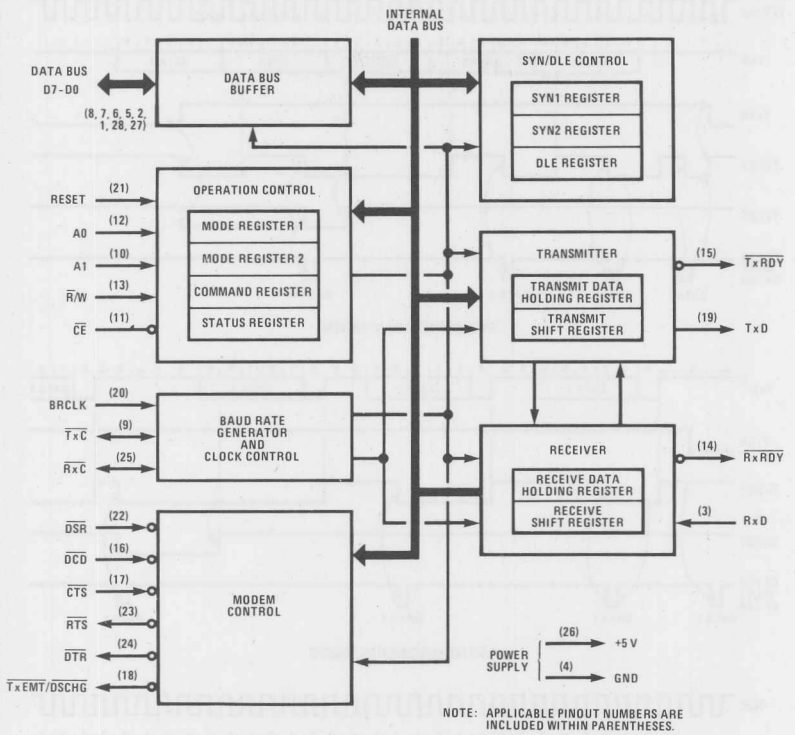


ASYNCHRONOUS MODE

TxRDY, TxEMT TIMING (SHOWN FOR 5-BIT CHARACTERS, NO PARITY, 2-STOP BITS (IN SYNCHRONOUS MODE).
RxRDY TIMING (SHOWN FOR 5-BIT CHARACTERS, NO PARITY, 2-STOP BITS (IN ASYNCHRONOUS).

NOTE 1: A – START BIT, B – STOP BIT 1, C – STOP BIT 2,
D – TxD MARKING CONDITION.

INS2651 Block Diagram



INS2651 Functional Pin Definitions

The following describes the function of all the INS2651 input/output pins. Some of these descriptions reference internal circuits.

INPUT SIGNALS

Reset (RESET), Pin 21: When high, performs a master reset on the INS2651. This signal asynchronously terminates any device activity and clears the Mode, Command, and Status Regsiters. The device assumes the idle state and remains in this mode until initialized with the appropriate control words.

Address Lines (A1-A0), Pins 10, 12: Address lines used to select internal Mode and Command registers.

Read/Write ($\overline{R}$/W), Pin 13: Controls the direction of data bus transfers. A high input allows data from the CPU to be loaded into the addressed register. A low input causes the contents of the addressed register to be present on the data bus.

Chip Enable ($\overline{CE}$), Pin 11: When low, indicates that control and data lines to the device are valid and that the specified operation should be performed. When high, places the device in the TRI-STATE® condition.

Baud Rate Generator Clock (BRCLK), Pin 20: 5.0688 MHz clock input to the internal Baud Rate Generator. Not required if external receiver and transmitter ($\overline{TxC}$ and $\overline{RxC}$) clocks are used.

Receiver Data (RxD), Pin 3: Serial data input to the receiver.

Data Set Ready ($\overline{DSR}$), Pin 22: General-purpose input which, when low, indicates either the Data Set Ready or Ring condition. Its complement is stored as Status Register bit 7. A change in state of this input causes a low output on $\overline{TXEMT/DSCHG}$.

Data Carrier Detect ($\overline{DCD}$), Pin 16: When low, enables the receiver to operate. The complement of this input is stored as Status Register bit 6, and an input change in state causes a low output on $\overline{TXEMT/DSCHG}$.

Clear to Send ($\overline{CTS}$), Pin 17: When low, enables the transmitter to operate. When high, holds the $\overline{TxD}$ output in MARK condition.

V_{CC}, Pin 26: +5-volt supply.

Ground, Pin 4: 0-volt reference.

OUTPUT SIGNALS

Transmitter Ready ($\overline{TxRDY}$), Pin 15: A low on this output, which is open-drain, indicates that Transmit Holding Register (THR) is ready to accept a data character from the CPU. This output, which is the complement of Status Register bit 0, goes high when the data character is loaded and is valid only when the transmitter is enabled. The $\overline{TxRDY}$ output can be used as an interrupt to the system.

Receiver Ready ($\overline{RxRDY}$), Pin 14: A low on this output, which is open-drain, indicates that the Receive Holding Register (RHR) has a character ready for input to the CPU. This output, which is the complement of Status Register bit 1, goes high either when the Receiver Holding Register is read by the CPU or when the receiver is disabled. The $\overline{RxRDY}$ output can be used as an interrupt to the system.

Transmitter Empty or Data Set Change ($\overline{TxEMT/DSCHG}$), Pin 18: A low on this output, which is open-drain, indicates that either the transmitter has completed serialization of the last character loaded by the CPU or that a change of state of the $\overline{DSR}$ or $\overline{DCD}$ inputs has occurred. If the TxEMT condition does not exist, this output goes high when the Status Register is read by the CPU. Otherwise, the Transmit Holding Register must be loaded by the CPU for this line to go high. The $\overline{TxEMT/DSCHG}$ output can be used as an interrupt to the system. This output is the complement of Status Register bit SR2.

Transmitter Data ($\overline{TxD}$), Pin 19: Composite serial data output to a MODEM or input/output device. The TxD output is held in the marking state (logic 1) when the transmitter is disabled.

Data Terminal Ready ($\overline{DTR}$), Pin 24: General-purpose output normally used to indicate Data Terminal Ready. The $\overline{DTR}$ output is the complement of Command Register bit 1.

Request to Send ($\overline{RTS}$), Pin 23: General-purpose output normally used to indicate Request to Send. The $\overline{RTS}$ output is the complement of Command Register bit 5.

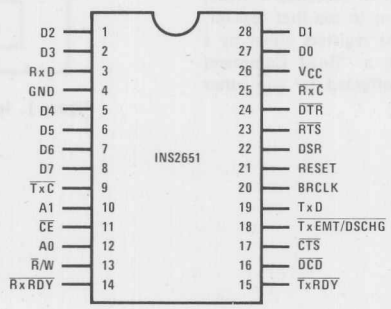
INPUT/OUTPUT SIGNALS

Data (D7-D0) Bus, Pins 28, 27, 8, 7, 6, 5, 2, 1: This bus comprises eight TRI-STATE input/output lines. The bus provides bidirectional communications between the INS2651 and the CPU. Data, control words, and status information are transferred via the Data Bus.

Receiver Clock ($\overline{RxC}$), Pin 25: If external receiver clock is programmed, this input controls the rate at which a data character is received. The frequency of the $\overline{RxC}$ input is a multiple (1x, 16x, or 64x) of the Baud Rate. Data is sampled on the rising edge of the clock. If internal receiver clock is programmed, this pin becomes an output at 1x the programmed Baud Rate.

Transmitter Clock ($\overline{TxC}$), Pin 9: If external transmitter clock is programmed, this input controls the rate at which a data character is transmitted. The frequency of the $\overline{TxC}$ input is a multiple (1x, 16x, or 64x) of the Baud Rate. Transmitter Data is clocked out of the INS2651 on the falling edge of the $\overline{TxC}$ input. If internal transmitter clock is programmed, this pin becomes an output at 1x the programmed Baud Rate.

Pin Configuration



INS2651 Programming

The system software determines the operative conditions (mode selection, clock selection, data format, and so forth) of the INS2651 via internal Mode Registers 1 and 2, and the Command Register. Prior to initiating data communications, the INS2651 operational mode must be programmed by performing write operations to these 8-bit registers via the Data Bus. The device can be reprogrammed at any time during program execution. However, the receiver and transmitter should be disabled if the change has an effect on the reception or transmission of a character.

The internal registers of the INS2651 are accessed by applying signals to the $\overline{CE}$, $\overline{R}/W$, A1, and A0 inputs as specified in table 1.

Table 1. Guess My Name

$\overline{CE}$	A1	A0	R/W	Function
1	X	X	X	TRI-STATE Data Bus
0	0	0	0	Read Receive Holding Register
0	0	0	1	Write Transmit Holding Register
0	0	1	0	Read Status Register
0	0	1	1	Write SYN1/SYN2/DLE Registers
0	1	0	0	Read Mode Registers 1 and 2
0	1	0	1	Write Mode Registers 1 and 2
0	1	1	0	Read Command Register
0	1	1	1	Write Command Register

In the case of multiple registers (SYN1/SYN2/DLE Registers and Mode Registers 1 and 2), successive read or write operations will access the next higher register. For example, if A1 equals 0, A2 equals 1, and $\overline{R}/W$ equals 1, the first write operation loads SYN1 Register. The next write operation loads SYN2 Register, and the third loads the DLE Register. Read and write operations are performed on the Mode Registers in a similar manner. If more than the required number of accessses is made, the internal register pointer returns to the first register. The pointers are reset to the first registers either by a RESET input or by performing a "Read Command Register" operation, but are unaffected by any other read or write operation.

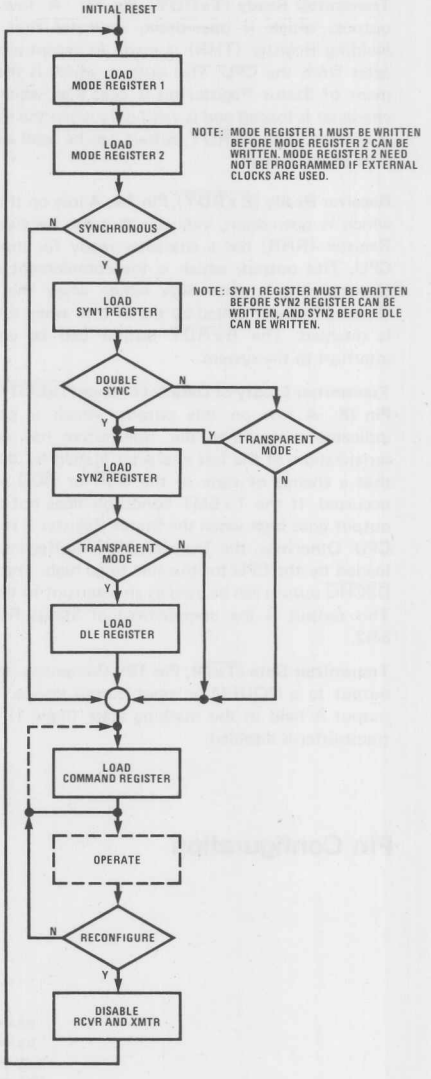


Figure 1. Initialization Flowchart

MODE REGISTER 1 FORMAT

BIT NUMBERS

MR1-7	MR1-6	MR1-5	MR1-4	MR1-3	MR1-2	MR1-1	MR1-0
SYNC: NO. OF SYN CHARACTERS 0 = DOUBLE SYN 1 = SINGLE SYN ASYNC: STOP BIT LENGTH 00 = INVALID 01 = 1 STOP BIT 10 = 1½ STOP BITS 11 = 2 STOP BITS	SYNC: TRANSPARENCY CONTROL 0 = NORMAL 1 = TRANSPARENT	PARITY TYPE 0 = ODD 1 = EVEN	PARITY CONTROL 0 = DISABLED 1 = ENABLED	CHARACTER LENGTH 00 = 5 BITS 01 = 6 BITS 10 = 7 BITS 11 = 8 BITS		MODE AND BAUD RATE FACTOR[1] 00 = SYNCHRONOUS 1x RATE 01 = ASYNCHRONOUS 1x RATE 10 = ASYNCHRONOUS 16x RATE 11 = ASYNCHRONOUS 64x RATE	

MODE REGISTER 2 FORMAT

BIT NUMBERS

MR2-7	MR1-6	MR2-5	MR2-4	MR2-3	MR2-2	MR2-1	MR2-0
NOT USED		TRANSMITTER CLOCK 0 = EXTERNAL 1 = INTERNAL	RECEIVER CLOCK 0 = EXTERNAL 1 = INTERNAL	BAUD RATE SELECTION 0000 = 50 BAUD 0001 = 75 BAUD 0010 = 110 BAUD 0011 = 134.5 BAUD 0100 = 150 BAUD 0101 = 300 BAUD	0110 = 600 BAUD 0111 = 1200 BAUD 1000 = 1800 BAUD 1001 = 2000 BAUD 1010 = 2400 BAUD 1011 = 3600 BAUD	1100 = 4800 BAUD 1101 = 7200 BAUD 1110 = 9600 BAUD 1111 = 19200 BAUD	

COMMAND REGISTER FORMAT

BIT NUMBERS

CR-7	CR-6	CR-5	CR-4	CR-3	CR-2	CR-1	CR-0
OPERATING MODE 00 = NORMAL OPERATION 01 = ASYNC: AUTOMATIC ECHO MODE SYNC: SYN AND/OR DLE STRIPPING MODE 10 = LOCAL LOOP BACK 11 = REMOTE LOOP BACK		REQUEST TO SEND 0 = FORCES $\overline{RTS}$ OUTPUT HIGH 1 = FORCES $\overline{RTS}$ OUTPUT LOW	RESET ERROR 0 = NORMAL 1 = RESET ERROR FLAG IN STATUS REGISTER (FE, OE, PE/DLE DETECT)	ASYNC: FORCE BREAK 0 = NORMAL 1 = FORCE BREAK SYNC: SEND DLE 0 = NORMAL 1 = SEND DLE	RECEIVE CONTROL (RxEN) 0 = DISABLE 1 = ENABLE	DATA TERMINAL READY 0 = FORCES $\overline{DTR}$ OUTPUT HIGH 1 = FORCES $\overline{DTR}$ OUTPUT LOW	TRANSMIT CONTROL 0 = DISABLE 1 = ENABLE

STATUS REGISTER FORMAT

BIT NUMBERS

SR-7	SR-6	SR-5	SR-4	SR-3	SR-2	SR-1	SR-0
DATA SET READY 0 = $\overline{DSR}$ INPUT IS HIGH 1 = $\overline{DSR}$ INPUT IS LOW	DATA CARRIER DETECT 0 = DCD INPUT IS HIGH 1 = DCD INPUT IS LOW	FE/SYN DETECT ASYN: 0 = NORMAL 1 = FRAMING ERROR SYNC: 0 = NORMAL 1 = SYN CHARACTER DETECTED	OVERRUN 0 = NORMAL 1 = OVERRUN ERROR	PE/DLE DETECT ASYNC: 0 = NORMAL 1 = PARITY ERROR SYNC: 0 = NORMAL 1 = PARITY ERROR OR DLE CHARACTER RECEIVED	TxEMT/DSCHG 0 = NORMAL 1 = CHANGE IN $\overline{DSR}$ OR $\overline{DCD}$, OR TRANSMIT SHIFT REGISTER IS EMPTY	RxRDY 0 = RECEIVE HOLDING REGISTER EMPTY 1 = RECEIVE HOLDING REGISTER HAS DATA	TxRDY 0 = TRANSMIT HOLDING REGISTER BUSY 1 = TRANSMIT HOLDING REGISTER EMPTY

NOTE 1: BAUD RATE FACTOR IN ASYNCHRONOUS MODE APPLIES ONLY IF EXTERNAL CLOCK IS SELECTED. FACTOR IS 16x IF INTERNAL CLOCK IS SELECTED.

Table 2. Baud Rate Generator Characteristics (Crystal Frequency = 5.0688 MHz)

Baud Rate	Theoretical Frequency 16x Clock (kHz)	Actual Frequency 16x Clock (kHz)	Percent Error	Duty Cycle (%)	Divisor
50	0.8	0.8	–	50/50	6336
75	1.2	1.2	–	50/50	4224
110	1.76	1.76	–	50/50	2880
134.5	2.152	2.1523	0.016	50/50	2355
150	2.4	2.4	–	50/50	2112
300	4.8	4.8	–	50/50	1056
600	9.6	9.6	–	50/50	528
1200	19.2	19.2	–	50/50	264
1800	28.8	28.8	–	50/50	176
2000	32.0	32.081	0.253	50/50	158
2400	38.4	38.4	–	50/50	132
3600	57.6	57.6	–	50/50	88
4800	76.8	76.8	–	50/50	66
7200	115.2	115.2	–	50/50	44
9600	153.6	153.6	–	48/52	33
19200	307.2	316.8	3.125	50/50	16

Note: 16x clock is used in asynchronous mode. In synchronous mode, clock multiplier is 1x and duty cycle is 50%/50% for any baud rate.

INS2651 Operation

GENERAL

The transmitter section of the INS2651 performs parallel-to-serial conversion of data supplied to it from the system data bus.

The receiver section of the INS2651 performs serial-to-parallel conversion of data received from the MODEM or input/output device. Both the transmitter and receiver are double buffered, allowing a full character time in which to service Transmit Ready ($\overline{TxRDY}$) and Receive Ready ($\overline{RxRDY}$) interrupts.

The character size (5, 6, 7, or 8 bits) is program selectable. Parity check/generation and the baud rate may also be defined by the program. Note that the character size is exclusive of the start/stop and parity bits.

SYNCHRONOUS MODE

The transmitter starts transmitting a continuous bit stream once the transmitter is enabled and the Clear to Send ($\overline{CTS}$) input is low. If the system is late in supplying a character to the transmitter, then the transmitter will send the SYN character (or SYN1, two characters if in double SYNC mode) as an idle fill in the Non-Transparent mode, or the DLE-SYN1 character pair as an idle fill in the Transparent mode. If this condition occurs, the $\overline{TxEMT/DSCHG}$ output goes low.

The receiver enters a character synchronization mode as soon as the receiver is enabled and the Data Carrier Detect ($\overline{DCD}$) input goes low. Either one or two consecutive SYN characters must be recognized by the receiver. The number of SYN characters is program selectable, and data is sent to the processor only after synchronization. The SYN character(s) in the Transparent mode (or DLE-SYN1 characters in the Non-Transparent mode) are stripped off the data stream after synchronization. This feature is program selectable.

An overrun error will occur if the processor is late in servicing the received character. When this condition occurs, the character in the receiver buffer is written over by the character causing the overrun, and the overrun status bit is set.

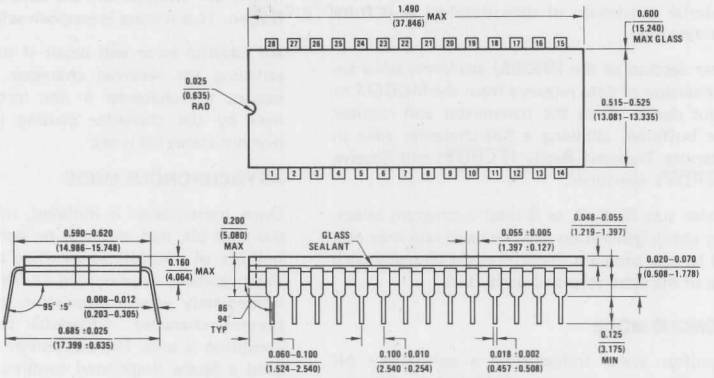
ASYNCHRONOUS MODE

Once transmission is initiated, the transmitter supplies the start bit, odd, even, or no parity bit, and the proper number of stop bits as specified by the program. If the next character is presented to the transmitter, it is sent immediately after transmission of the stop bit of the present character. Otherwise the Mark (logic high) condition is sent. The transmitter can be programmed to send a Space (logic low) condition instead of the Mark condition.

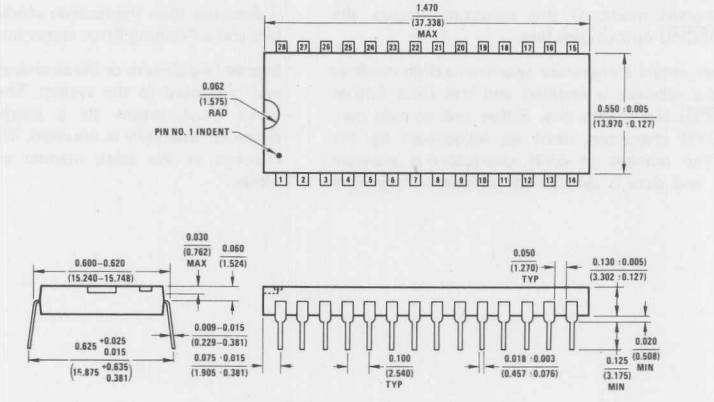
Once the receiver is enabled, reception of a character is initiated by recognition of the start bit. The Start/Stop and Parity bits are stripped off while assembling the serial input into a parallel character. If a break condition is detected then the receiver sends a character of all zero bits and a Framing Error status bit to the processor.

Succeeding all-zero or break characters are not assembled and presented to the system. The Receive Data (RxD) input must return to a marking condition before character assembly is resumed. The overrun condition is checked in the same manner as in the Synchronous mode.

Physical Dimensions



28-Lead Ceramic Dual-In-Line Package [Cer Dip (J)]
Order Number INS2651J



28-Lead Plastic Dual-In-Line Package (N)
Order Number INS2651N

National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, California 95051
Tel.: (408) 737-5000
TWX: (910) 339-9240

National Semiconductor GmbH
Eisenheimerstrasse 61/II
8000 Munchen 21
West Germany
Tel.: 089/9 15027
Telex: 05-22772

NS International Inc., Japan
Miyake Building
1-9 Yotsuya, Shinjuku-ku 160
Tokyo, Japan
Tel.: (03) 355-3711
TWX: 232-2015 NSCJ-J

National Semiconductor (Hong Kong) Ltd.
8th Floor,
Cheung Kong Electronic Bldg.
4 Hing Yip Street
Kwun Tong
Kowloon, Hong Kong
Tel.: 3-411241-8
Telex: 43866 NSEHK HX
Cable: NATSEMI

NS Electronics Do Brasil
Avda Brigadeiro Faria Lima 844
11 Andar Conjunto 1104
Jardim Paulistano
Sao Paulo, Brasil
Telex:
1121008 CABINE SAO PAULO

NS Electronics Pty. Ltd.
Cnr. Stud Rd. & Mtn. Highway
Bayswater, Victoria 3153
Australia
Tel.: 03-729-6333
Telex: 32096

National Semiconductor

Preliminary
JULY 1978

INS8250 Asynchronous Communications Element

General Description

The INS8250 is a programmable Asynchronous Communications Element (ACE) chip contained in a standard 40-pin dual-in-line package. The chip, which is fabricated using N-channel silicon gate technology, functions as a serial data input/output interface in a microcomputer system. The functional configuration of the INS8250 is programmed by the system software via a TRI-STATE® 8-bit bidirectional data bus.

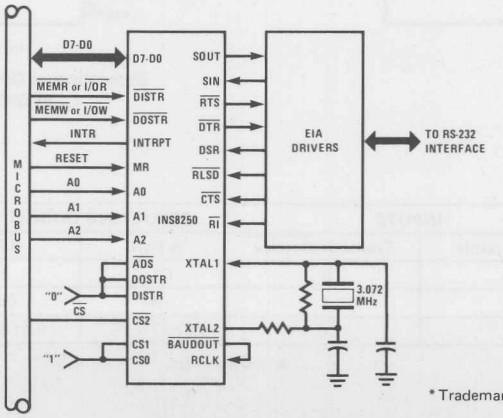
The INS8250 performs serial-to-parallel conversion on data characters received from a peripheral device or a MODEM, and parallel-to-serial conversion on data characters received from the CPU. The CPU can read the complete status of the INS8250 at any time during the functional operation. Status information reported includes the type and condition of the transfer operations being performed by the INS8250, as well as any error conditions (parity, overrun, framing, or break interrupt).

In addition to providing control of asynchronous data communications, the INS8250 includes a programmable Baud Generator that is capable of dividing the timing reference clock input by divisors of 1 to $(2^{16} - 1)$, and producing a 16x clock for driving the internal transmitter logic. Provisions are also included to use this 16x clock to drive the receiver logic. Also included in the INS8250 is a complete MODEM-control capability, and a processor-interrupt system that may be software tailored to the user's requirements to minimize the computing time required to handle the communications link.

Features

- Designed to be Easily Interfaced to Most Popular Microprocessors.
- Adds or Deletes Standard Asynchronous Communication Bits (Start, Stop, and Parity) to or from Serial Data Stream
- Full Double Buffering Eliminates Need for Precise Synchronization
- Independently Controlled Transmit, Receive, Line Status, and Data Set Interrupts
- Programmable Baud Rate Generator Allows Division of Any Input Clock by 1 to $(2^{16} - 1)$ and Generates the Internal 16x Clock
- Independent Receiver Clock Input
- MODEM Control Functions (CTS, RTS, DSR, DTR, RI, and Carrier Detect)
- Fully Programmable Serial-Interface Characteristics
  - 5-, 6-, 7-, or 8-Bit Characters
  - Even, Odd, or No-Parity Bit Generation and Detection
  - 1-, 1½-, or 2-Stop Bit Generation
  - Baud Rate Generation (DC to 56k Baud)
- False Start Bit Detection
- Complete Status Reporting Capabilities
- TRI-STATE TTL Drive Capabilities for Bidirectional Data Bus and Control Bus
- Line Break Generation and Detection
- Internal Diagnostic Capabilities
  - Loopback Controls for Communications Link Fault Isolation
  - Break, Parity, Overrun, Framing Error Simulation
- Full Prioritized Interrupt System Controls
- Single +5-Volt Power Supply
- MICROBUS™* Compatible

INS8250 MICROBUS Configuration



*Trademark, National Semiconductor Corp.

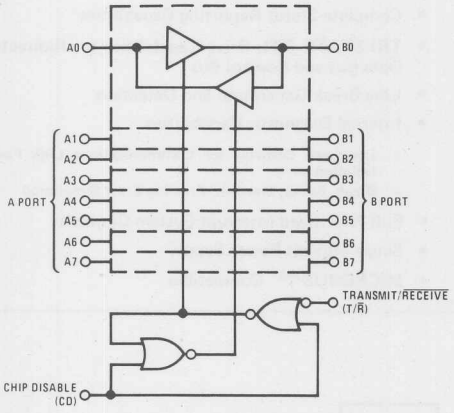
National Semiconductor

DP7304B/DP8304B 8-Bit TRI-STATE® Bidirectional Transceiver (Non-Inverting)

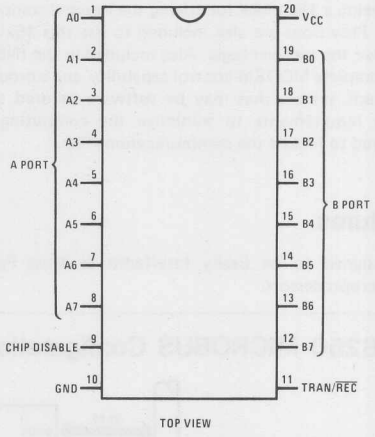
Features

- 8-bit bidirectional data flow reduces system package count
- Bidirectional TRI-STATE inputs/outputs interface with bus oriented systems
- PNP inputs reduce input loading
- Output high voltage interfaces with TTL, MOS, and CMOS
- 48 mA/300 pF bus drive capability
- Pinouts simplify system interconnections
- Transmit/Receive and chip disable simplify control logic
- Compact 20-pin dual-in-line package
- Bus port glitch free power up/down

Logic and Connection Diagrams



Dual-In-Line Package



TOP VIEW

Order Number DP7304BJ, DP8304BJ or DP8304BN

Logic Table

INPUTS		RESULTING CONDITIONS	
Chip Disable	Transmit/Receive	A Port	B Port
0	0	OUT	IN
0	1	IN	OUT
1	X	TRI-STATE	TRI-STATE

X = Don't care

Digital I/O Components

National Semiconductor

Preliminary
July 1979

INS8243 Input/Output Expander

General Description

The INS8243 is an input/output device specifically designed to furnish input/output expansion capabilities for the INS8048, INS8049 and INS8050 single chip microcomputer family. The INS8243 is fabricated using XMOS (high density N-channel silicon gate) technology, operates from a single 5 volt supply and is TTL compatible. It is housed in a 24-pin, dual-in-line package and provides high drive current capabilities at low cost.

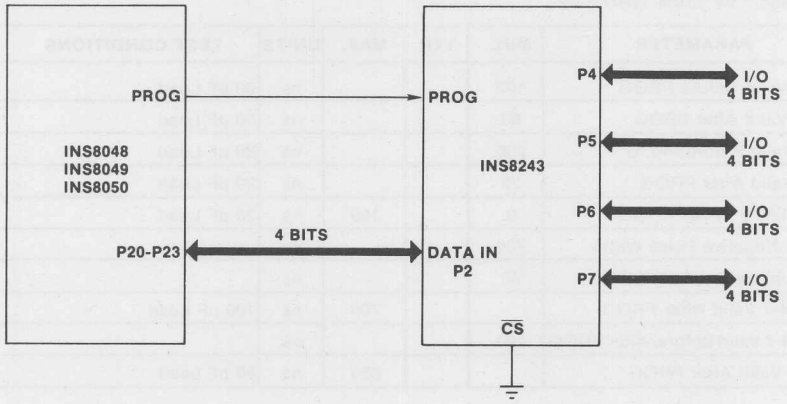
The INS8243 expander consists of five, 4-bit bidirectional ports. One port provides the interface with the INS8048/49/50 microcomputer. The remaining four ports provide the input/output expansion.

The INS8243 I/O ports function as a direct extension for the resident I/O port of the INS8048/49/50 microcomputer series and are accessed by the MOV, ANL or ORL instructions of the INS8048/49/50.

Features

- XMOS Technology
- Single 5V Supply
- Low Cost I/O Expansion
- Easy Interface with INS8048/49/50 Microcomputers
- High Fanout Capability
- 24-Pin DIP
- Direct Extension of INS8048/49/50 I/O Ports

INS8243 Basic System Configuration



Single Expander Interface

Absolute Maximum Ratings

Ambient Temperature Under Bias ..... 0°C to +70°C
Storage Temperature Range..... -65°C to +150°C
Voltage on Any Pin with Respect to GND ..... -0.5V to +7.0V

NOTE: *Absolute maximum ratings indicate limits beyond which permanent damage may occur. Continuous operation at these limits is not intended; operation should be limited to those conditions specified under DC Electrical Characteristics.*

DC Electrical Characteristics

T_A = 0°C to +70°C, V_{CC} = +5V ±10%, GND = 0V

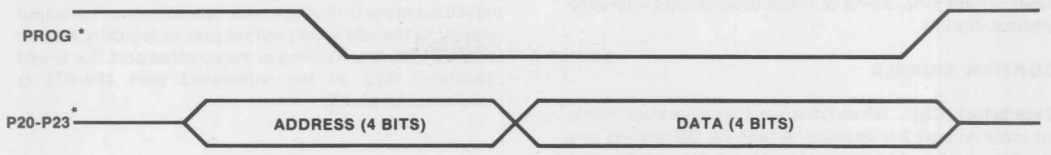
SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNITS	TEST CONDITIONS
V_{IL}	Input Low Voltage	-0.5		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC}+0.5	V	
V_{OL1}	Output Low Voltage Ports 4-7			0.45	V	I_{OL} = 5 mA
V_{OL2}	Output Low Voltage Port 7			1	V	I_{OL} = 20 mA
V_{OH1}	Output High Voltage Ports 4-7	2.4			V	I_{OH} = 240 μA
I_{IL1}	Input Leakage Ports 4-7	-10		20	μA	V_{IN} = V_{CC} to 0V
I_{IL2}	Input Leakage Port 2, $\overline{CS}$, PROG	-10		10	μA	V_{IN} = V_{CC} to 0V
V_{OL3}	Output Low Voltage Port 2			0.45	V	I_{OL} = 0.6 mA
I_{CC}	V_{CC} Supply Current		10	20	mA	
V_{OH2}	Output Voltage Port 2	2.4			V	I_{OH} = 100 μA
I_{OL}	Sum of all I_{OL} from 16 output ports			100	mA	At 5 mA per pin

AC Characteristics

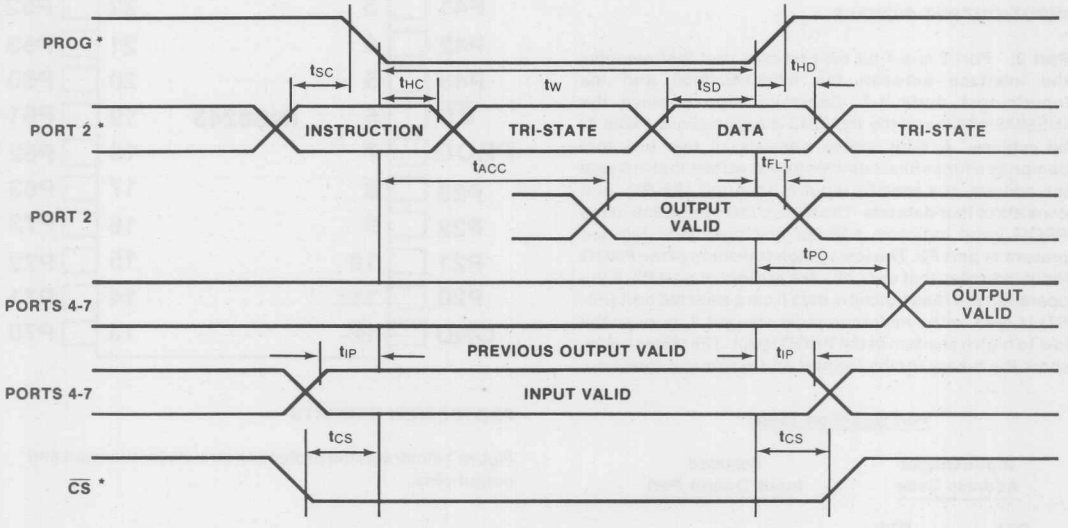
T_A = 0°C to 70°C, V_{CC} = 5V ±10%, GND = 0V

SYMBOL	PARAMETER	MIN.	TYP	MAX.	UNITS	TEST CONDITIONS
t_{SC}	Code Valid Before PROG	100			ns	80 pF Load
t_{HC}	Code Valid After PROG	60			ns	20 pF Load
t_{SD}	Data Valid Before PROG	200			ns	80 pF Load
t_{HD}	Data Valid After PROG	20			ns	20 pF Load
t_{FLT}	TRI-STATE™ After PROG	0		150	ns	20 pF Load
t_W	PROG Negative Pulse Width	700			ns	
t_{CS}	$\overline{CS}$ Valid Before/After PROG	50			ns	
t_{PO}	Ports 4-7 Valid After PROG			700	ns	100 pF Load
t_{LP1}	Ports 4-7 Valid Before/After PROG	100			ns	
t_{ACC}	Port 2 Valid After PROG			650	ns	80 pF Load

Timing Waveforms (*These signals are generated by the INS8048/49/50.)



Expander Input Timing



Input/Output Waveforms

Functional Pin Definitions

The following describes the function of the INS8243 input/output pins. Some of these descriptions reference internal circuits.

CONTROL SIGNALS

Chip Select ($\overline{CS}$): When $\overline{CS}$ is low (negative true), the 4-bit input to port 2 is enabled. A high on $\overline{CS}$ inhibits any input to port 2 and no change to internal status and output can occur.

Strobe Input (PROG): The low to high transition on the PROG indicates data is available at port 2; a high to low transition on PROG signifies command and address information is at port 2.

INPUT/OUTPUT SIGNALS

Port 2: Port 2 is a 4-bit bidirectional port that provides the interface between the INS8048/49/50 and the input/output ports 4-7. Communication between the INS8048/49/50 and the INS8243 is accomplished with 4-bit nibbles. A 4-bit nibble consists of two bits that comprise a functional command and two bits that indicate the address of a specific input/output port (P4-P7), or it consists of four data bits. The high to low transistion of the PROG input indicates address and command bits are present at port P2. The low to high transition of the PROG input indicates that data bits are present at port P2. If the operation is a read function, data from a selected port (P4-P7) is read to the microcomputer via port 2, prior to the low to high transition of the PROG input. The tables below show the binary inputs for port and command selection.

Port Selection Table

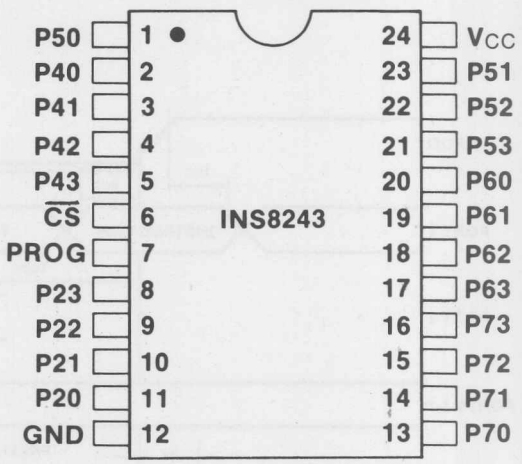
Input/Output Address Code		Selected Input/Output Port
P21	P20	
0	0	4
0	1	5
1	0	6
1	1	7

Functional Command Selection Table

Command Code		Function
P23	P22	
0	0	Read
0	1	Write
1	0	ORLD
1	1	ANLD

Ports 4-7 (Pi): These are four, 4-bit bidirectional input/output ports. Each port is addressable and may be programmed to perform a read (input data) or write (output data) via a low impedance latched output. Data presented to port 2 during a write operation may be output directly to the addressed output port or logically ANDed or ORed with data existing in the selected port. For a read operation, data at the addressed port (P4-P7) is transferred to port 2.

Pin Configuration



PROTECTION CIRCUITS

Figure 1 illustrates the protection circuits for the input and output pins.

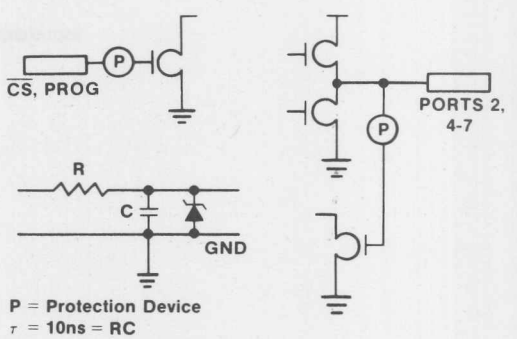


FIGURE 1. Protection Circuits for I/O Pins

Functional Description

The INS8243 (*Figure 2*) consists of four, 4-bit I/O ports that function as the extension of the INS8048/49/50 on-chip input/output port. The four I/O ports (P4 through P7) are addressable and the following programmed INS8048/49/50 instructions are used to access these ports. The instructions move data to/from the INS8048/49/50 accumulator via the INS8243. Timing for the transfer of data is provided by the INS8048/49/50 PROG output.

a. MOVD Pi,A - Shift accumulator data to the addressed port.

b. MOVD A,Pi - Shift addressed port data to the accumulator.

c. ANLD Pi,A - ANDing accumulator data to addressed port.

d. ORLD Pi,A - ORing accumulator data to addressed port.

Port 2 of the INS8243 provides the communication interface between the expander and the INS8048/49/50 processor. Each communications exchange is comprised of two, 4-bit nibbles, one nibble consisting of command and address information, the second is a 4-bit data nibble. Timing for the I/O expander is provided by the processor on the PROG input pin.

Power-On Initializing Mode

Application of DC power to the chip forces I/O port 2 to the input mode and I/O ports 4 through 7 to the TRI-STATE® output mode (high impedance state). The power-on sequence is initiated when V_{CC} falls below 1 volt. The input level on the PROG may be high or low when DC is first applied. The initial high to low transition of the PROG input forces the chip to exit the power-on mode.

Read Mode

The INS8243 I/O expander has one read mode that is initialized by the following instruction:

- MOVDA,Pi - Instruction from INS8048/49/50 takes data from the addressed I/O port (P4-P7) and moves the data into the accumulator.

The command code and port address code are latched from the input port 2 on the high to low transition of the PROG input pin. When the read command and the port address are decoded, the data from the addressed port is presented back to the INS8048/49/50 on port P2. Termination of the read command occurs on the low to high transition of the PROG input. The port (4, 5, 6 or 7) that was addressed switches to the high impedance state and port 2 reverts to the input (read) mode. A port will normally be in either write (output) mode or read (input) mode. To allow for the settling of the external driver on the port, the first read following a write should be discounted when modes are changed during operation. All succeeding reads are valid.

Write Modes

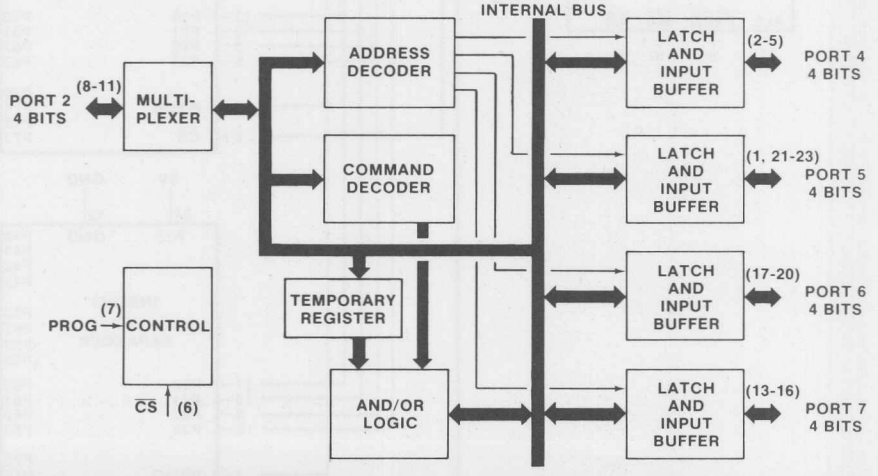
The INS8243 has three write modes that are initialized by the following instructions:

a. MOVD Pi,A - Instruction from the INS8048/49/50 writes new data directly into the addressed port. Existing data is lost.

b. ORLD Pi,A - Instruction from the INS8048/49/50 takes port P2 data, logically ORing it with the existing data in the addressed port and writes the resultant data into the port.

c. ANLD Pi,A - Instruction from the INS8048/49/50 takes port P2 data, logically ANDing it with the existing data in the addressed port and writes the ANDed data into the port.

Command and port address codes are latched from the port 2 input on the high to low transition of the PROG input. Data on port 2 is deposited in the logic circuits of the addressed port. When the logic manipulation has been performed, data is latched and output. Old data stays latched until valid new outputs are written.



Note: Applicable pinout numbers are included within parentheses

FIGURE 2. INS8243 I/O Expander Block Diagram

Multiple INS8243 I/O Expander Use

To expand the resident I/O port capabilities of the INS8048/49/50 microcomputer series sixteen times, four INS8243 expanders may be connected to the microcomputer, as shown in *Figure 3*. By using the high order bit outputs P24 through P27 as the chip select ($\overline{CS}$) input to the I/O expanders, no additional logic circuits are required. A negative true input (low) on the $\overline{CS}$ line will enable the port 2 input for the selected I/O expander. A single INS8048/49/50 microcomputer may use many (up to twenty) I/O expanders without additional logic circuits on the same bus, limited only by the availability of chip select lines and loading restrictions.

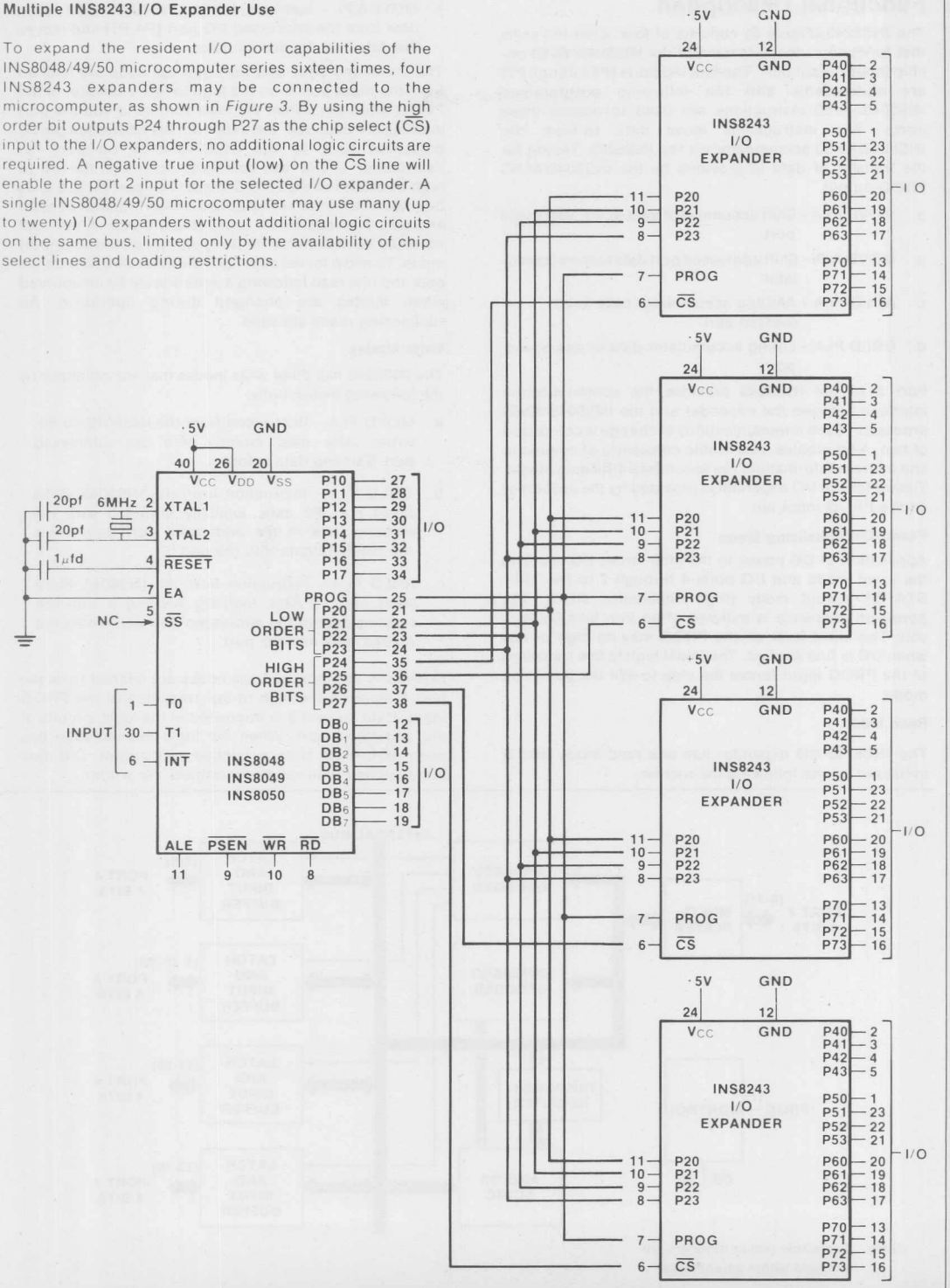


FIGURE 3. Typical Connections Using INS8243 I/O Expander

Larger numbers of INS8243 expanders would require chip select decoder chips to conserve microcomputer I/O pins. Operation of the INS8243 expander selected is the same as was explained in the functional description. *Figure 4* is a typical system application.

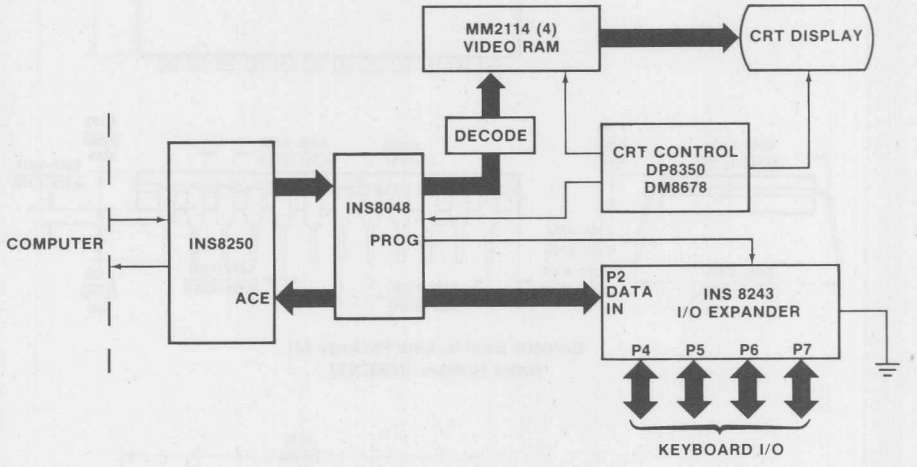
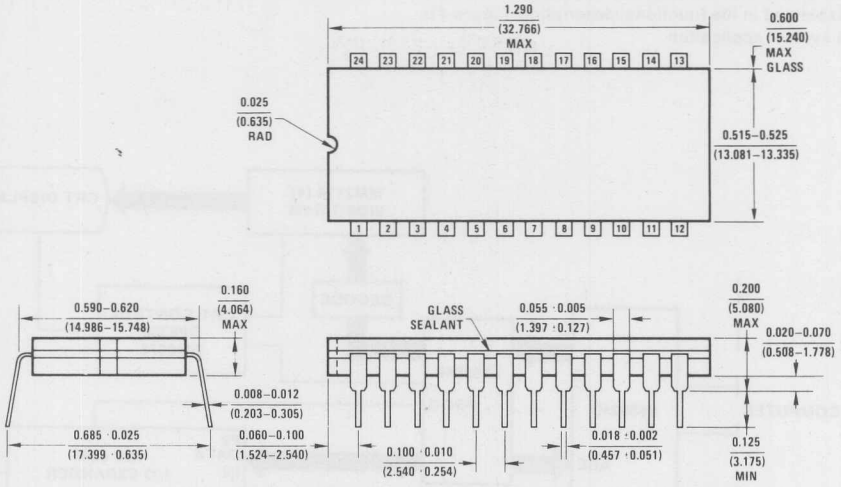


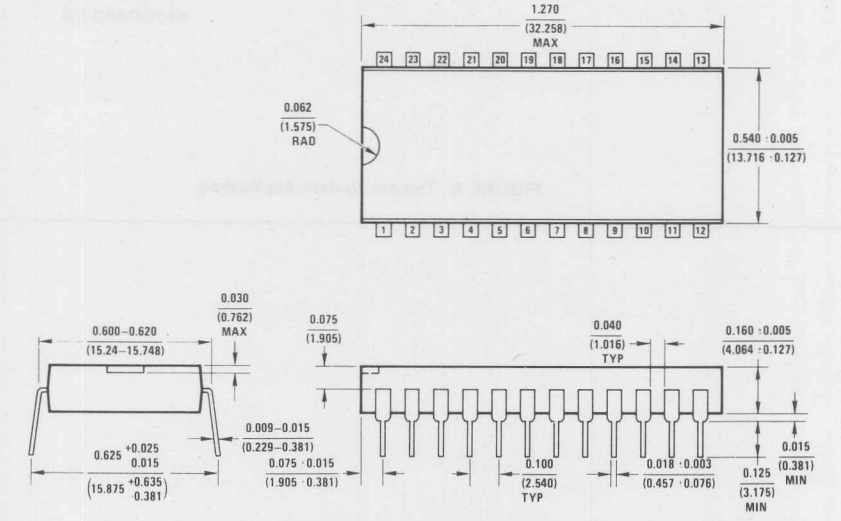
FIGURE 4. Typical System Application

Physical Dimensions

inches (millimeters)



Ceramic Dual-in-Line Package (J)
Order Number INS8243J



Plastic Dual-in-Line Package (N)
Order Number INS8243N

National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, California 95051
Tel (408) 737-5000
TWX (910) 339-9240

National Semiconductor GmbH
Eisenheimerstrasse 61/II
8000 Munchen 21
West Germany
Tel (089) 576091
Telex 05-22772

NS International Inc., Japan
Miyake Building
1-9 Yotsuya, Shinjuku-ku 160
Tokyo, Japan
Tel (03) 355-3711
TWX 232-2015 NSCJ-J

National Semiconductor (Hong Kong) Ltd.
8th Floor,
Cheung Kong Electronic Bldg
4 Hing Yip Street
Kwun Tong
Kowloon, Hong Kong
Tel 3-899235
Telex 73866 NSEHK HX
Cable NATSEMI

NS Electronics Do Brasil
Avda Brigadeiro Faria Lima 844
11 Andar Conjunto 1104
Jardim Paulistano
Sao Paulo, Brasil
Telex
1121008 CABINE SAO PAULO

NS Electronics Pty. Ltd.
Cnr. Stud Rd. & Mtn. Highway
Bayswater, Victoria 3153
Australia
Tel 03-729-6333
Telex 32096

National Semiconductor

Applicable TTL, ECL and CMOS Logic Circuits

DM7131/DM8131, DM7136/DM8136 6-Bit Unified Bus Comparators

General Description

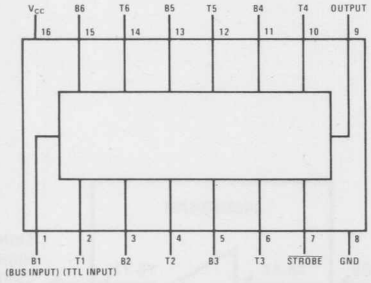
The DM7131/DM8131, DM7136/DM8136 compare two binary words of two-to-six-bits in length, and indicates matching (bit-for-bit) of the two words. Inputs for one word are 54/74 series-compatible TTL inputs, whereas those of the second word are high-impedance receivers driven by a terminated data bus. These bus inputs include 0.65V typical hysteresis, which provides 1.4V noise immunity. The DM7131/DM8131 has active pull-up outputs and goes to the low state upon equality. The DM7136/DM8136 has open-collector outputs which go to the high state upon equality, and is expandable to n bits by collector-ORing. Both devices have an output latch which is strobe controlled.

The transfer of information to the output occurs when the $\overline{\text{STROBE}}$ input goes from a logic "1" to a logic "0" state. Inputs may be changed while the $\overline{\text{STROBE}}$ is at the logic "1" level, without affecting the state of the output. These devices are useful as address comparators in computer systems utilizing unified data bus organization.

Features

- Low bus input current 15μA typ
- High bus input noise immunity 1.4 typ
- Bus inputs comply with IEEE 488-1975
- TTL-compatible output
- Output latch provision

Connection Diagram



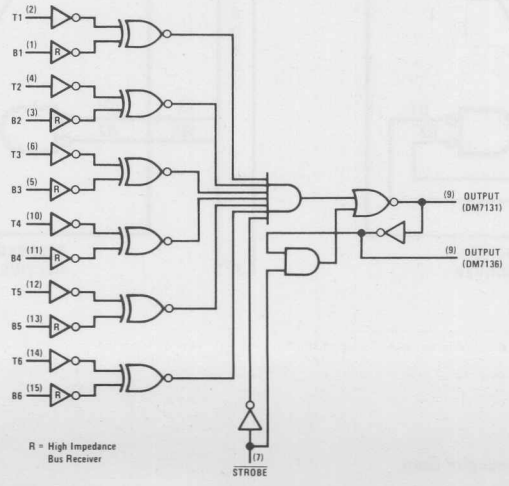
7131(J), (W); 8131(J), (N), (W); 7136(J), (W); 8136(J), (N), (W)

Truth Table

CONDITION	$\overline{\text{STROBE}}$	OUTPUT	
		DM71/8131	DM71/8136
T = B, T ≠ B	H	Q_{N-1} *	Q_{N-1} *
T = B	L	L	H
T ≠ B	L	H	L

*Latched in previous state

Logic Diagram



National Semiconductor

JULY 1978

INS8202/8203 TRI-STATE® Octal Buffers

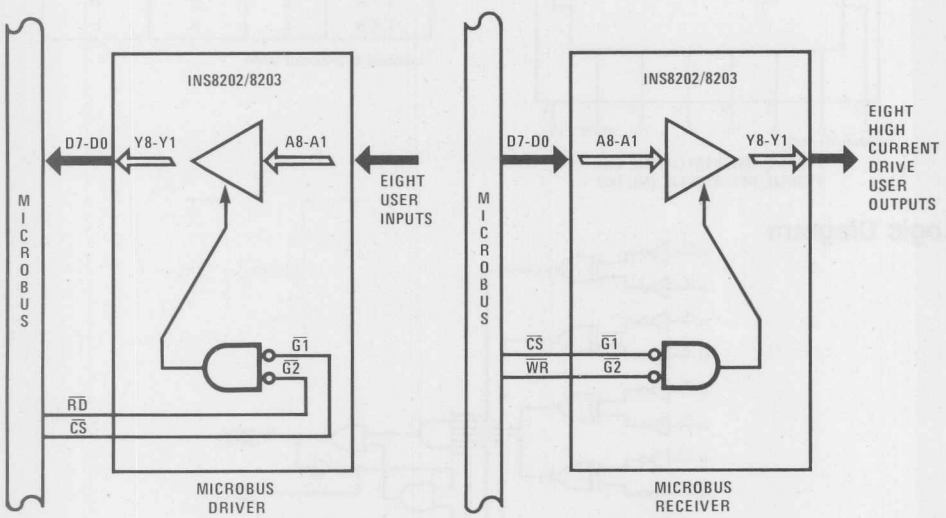
General Description

These devices provide eight two-input buffers in each package. All employ the newest low power Schottky TTL technology. One of the two inputs to each buffer is used as a control line to gate the output into the high-impedance state, while the other input passes the data through the buffer. The INS8202 presents true data at the outputs, while the INS8203 is inverting. All eight TRI-STATE enable lines are common, with access through a 2-input NOR gate. The outputs are placed in the TRI-STATE condition by applying a high logic level to the enable pins. These devices represent octal, low power Schottky versions of the very popular DM8095 and 96 TRI-STATE hex buffers.

Features

- Identical to DM81LS95 and DM81LS96
- Octal versions of popular DM80LS95, 80LS96
- Typical power dissipation
  INS8202 — 80mW
  INS8203 — 65mW
- Typical propagation delay
  INS8202 — 13ns
  INS8203 — 10ns
- Low power Schottky TRI-STATE technology
- MICROBUS™* compatible

INS8202/8203 MICROBUS Configuration



*Trademark, National Semiconductor Corp.

National Semiconductor

APRIL 1978

INS8208 8-Bit Bidirectional Transceiver

General Description

The INS8208 is an 8-bit TRI-STATE® low power Schottky transceiver. It provides bidirectional drive for bus-oriented microprocessor and digital communications systems. Straight through bidirectional transceivers are featured, with low power Schottky drive capability on the A ports and 48mA bus drive capability on the B ports. PNP inputs are incorporated to reduce input loading.

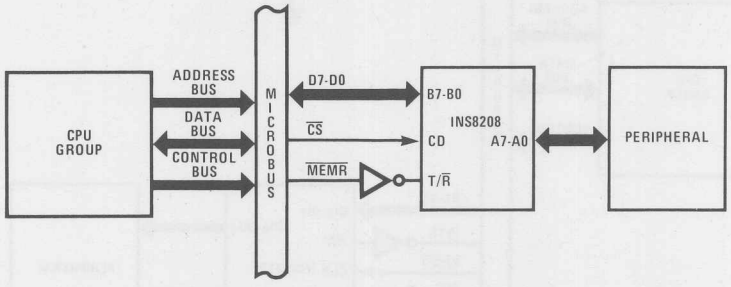
One input, Transmit/Receive, determines the direction of logic signals through the bidirectional transceiver: Transmit enables data from A ports to B ports; Receive enables data from B ports to A ports. The Chip Disable input disables both A and B ports by placing them in a TRI-STATE® condition.

The output high voltage (V_{OH}) is specified at 3.6 V minimum to allow interfacing microprocessors, T$^{\sim}$L, MOS, CMOS, RAM, or ROM.

Features

- 8-Bit Bidirectional Data Flow Reduces System Package Count
- Bidirectional TRI-STATE® Inputs/Outputs Interface with Bus-Oriented Systems
- PNP inputs Reduce Input Loading
- 3.6V Output High Voltage Interfaces with TTL, MOS, and CMOS
- 48mA/300pF Bus Drive Capability
- Pinouts Simplify System Interconnections
- Transmit/Receive and Chip Disable Simplify Control Logic
- Compact 20-Pin Dual-In-Line Package
- Low I_{CC} Power (8mA per bidirectional bit)
- MICROBUS™ * Compatible

INS8208 MICROBUS™ Configuration



* Trademark, National Semiconductor Corporation

National Semiconductor

APRIL 1978

INS8212 8-Bit Input/Output Port

General Description

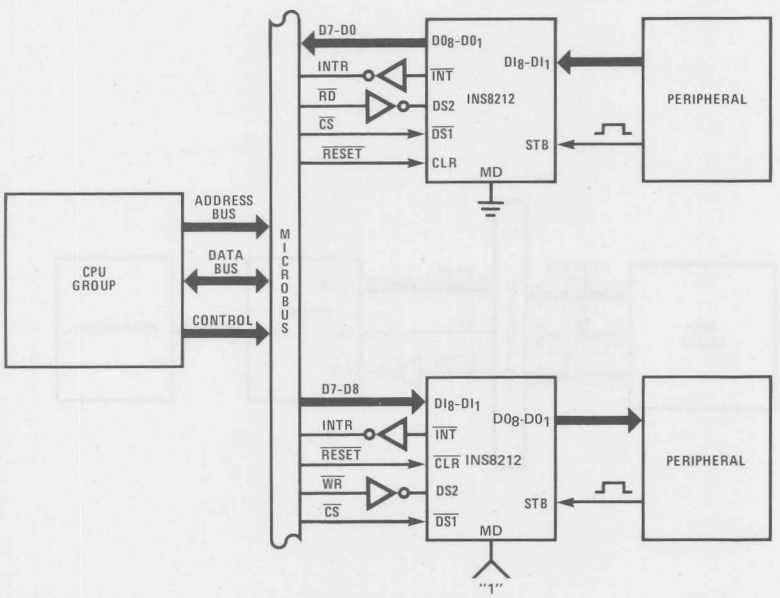
The INS8212 is an 8-bit input/output port contained in a standard 24-pin dual-in-line package. The device, which is fabricated using Schottky Bipolar technology, is part of National Semiconductor's INS8080A microprocessor family. The INS8212 can be used to implement latches, gated buffers, or multiplexers. Thus, all of the major peripheral and input/output functions of a microcomputer system can be implemented with this device.

The INS8212 includes an 8-bit latch with TRI-STATE® output buffers, and device selection and control logic. Also included is a service request flip-flop for the generation and control of interrupts to the microprocessor.

Features

- 8-Bit Data Latch and Buffer
- Service Request Flip-Flop for Generation and Control of Interrupts
- 0.25 mA Input Load Current
- TRI-STATE TTL Output Drive Capability
- Outputs Sink 15 mA
- Asynchronous Latch Clear
- 3.65V Output for Direct Interface to INS8080A
- Reduces System Package Count by Replacing Buffers, Latches, and Multiplexers in Microcomputer Systems
- MICROBUS™* Compatible

INS8212 MICROBUS Configuration



*Trademark, National Semiconductor Corp.

AUGUST 1978

National Semiconductor

INS8216/8226 4-Bit Bidirectional Bus Transceivers

General Description

The INS8216 and INS8226 are four-bit bidirectional bus drivers for use in bus oriented applications. The non-inverting INS8216 and inverting INS8226 drivers are provided for flexibility in system design.

Each buffered line of the four-bit driver consists of two separate buffers that are TRI-STATE® to achieve direct bus interface and bidirectional capability. On one side of the driver the output of one buffer and the input of another are tied together (DB); this side is used to interface to the system side components such as memories, I/O, etc., because its interface is TTL compatible and it has high drive (50mA). On the other side of the driver the inputs and outputs are separated to provide maximum flexibility. Of course, they can be tied together so that the driver can be used to buffer a true bidirectional bus. The DO outputs on this side of the driver have a special high voltage output drive capability so that direct interface to the 8080 type CPUs is achieved with an adequate amount of noise immunity.

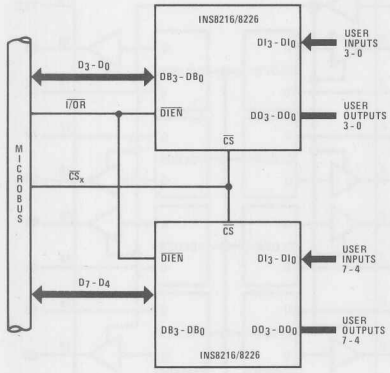
The $\overline{CS}$ input is a device enable. When it is "high" the output drivers are all forced to their high-impedance state. When it is a "low" the device is enabled and the direction of the data flow is determined by the $\overline{DIEN}$ input.

The $\overline{DIEN}$ input controls the direction of data flow, which is accomplished by forcing one of the pair of buffers into its high-impedance state and allowing the other to transmit its data. A simple two-gate circuit is used for this function.

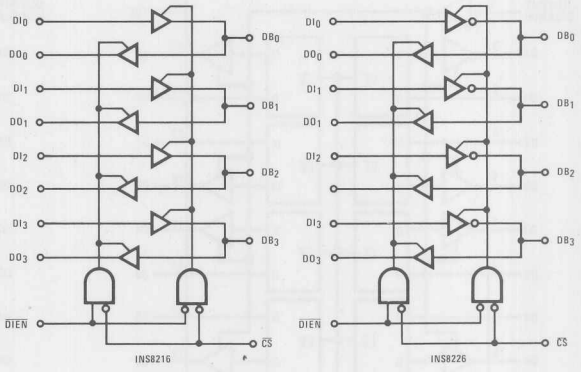
Features

- Data bus buffer driver for 8080 type CPUs
- Low input load current — 0.25mA maximum
- High output drive capability for driving system data bus — 50mA at 0.5V
- Power up-down protection
- The INS8216 has non-inverting outputs.
- The INS8226 has inverting outputs.
- Output high voltage compatible with direct interface to MOS
- TRI-STATE outputs
- Advanced Schottky processing
- Available in military and commercial temperature ranges
- MICROBUS™* compatible

INS8216/8226 MICROBUS Configuration



Logic Diagrams



*Trademark, National Semiconductor Corp.

National Semiconductor

Applicable TTL, ECL and CMOS Logic Circuits

MM54C373/MM74C373 TRI-STATE® Octal D-Type Latch
MM54C374/MM74C374 TRI-STATE® Octal D-Type Flip-Flop

General Description

The MM54C373/MM74C373, MM54C374/MM74C374 are integrated, complementary MOS (CMOS), 8-bit storage elements with TRI-STATE® outputs. These outputs have been specially designed to drive highly capacitive loads, such as one might find when driving a bus, and to have a fan-out of 1 when driving standard TTL. When a high logic level is applied to the OUTPUT DISABLE input, all outputs go to a high impedance state, regardless of what signals are present at the other inputs and the state of the storage elements.

The MM54C373/MM74C373 is an 8-bit latch. When $\overline{\text{LATCH ENABLE}}$ is high the Q outputs will follow the D inputs. When $\overline{\text{LATCH ENABLE}}$ goes low, data at the D inputs, which meets the set-up and hold time requirements, will be retained at the outputs until $\overline{\text{LATCH ENABLE}}$ returns high again.

The MM54C374/MM74C374 is an 8-bit, D-type, positive-edge triggered flip-flop. Data at the D inputs, meeting the set-up and hold time requirements, is transferred to the Q outputs on positive-going transitions of the CLOCK input.

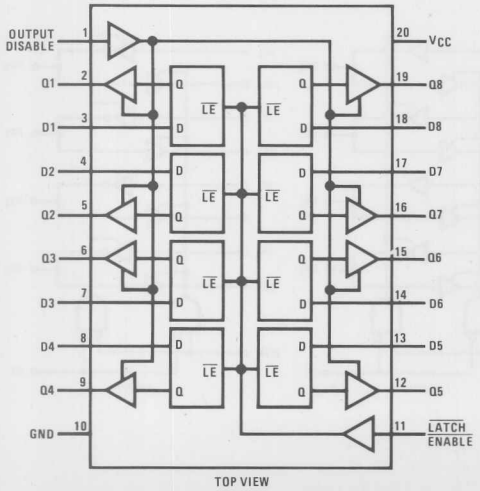
Both the MM54C373/MM74C373 and the MM54C374/MM74C374 are being assembled in 20-pin dual-in-line packages with 0.300" pin centers.

Features

- Wide supply voltage range — 3.0V to 15V
- High noise immunity — 0.45 V_{CC} typ
- Low power consumption
- TTL compatibility — fan-out of 1 driving standard TTL
- Bus driving capability
- TRI-STATE outputs
- Eight storage elements in one package
- Single CLOCK/$\overline{\text{LATCH ENABLE}}$ and OUTPUT DISABLE control inputs
- 20-pin dual-in-line package with 0.300" centers takes half the board space of a 24-pin package

Connection Diagrams

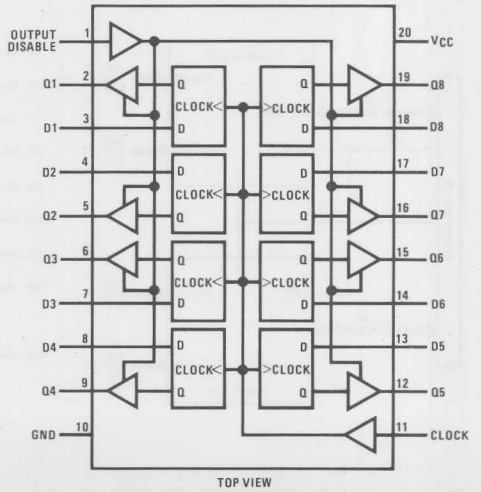
Dual-In-Line Package



TOP VIEW

Order Number MM54C373J or MM74C373N
See NS Package J20A or N20A

Dual-In-Line Package



TOP VIEW

Order Number MM54C374J or MM74C374N
See NS Package J20A or N20A

Memory Components

National Semiconductor

November 1979

MM2716 16,384-Bit (2048 × 8) UV Erasable PROM

General Description

The MM2716 is a high speed 16k UV erasable and electrically reprogrammable EPROM ideally suited for applications where fast turn-around and pattern experimentation are important requirements.

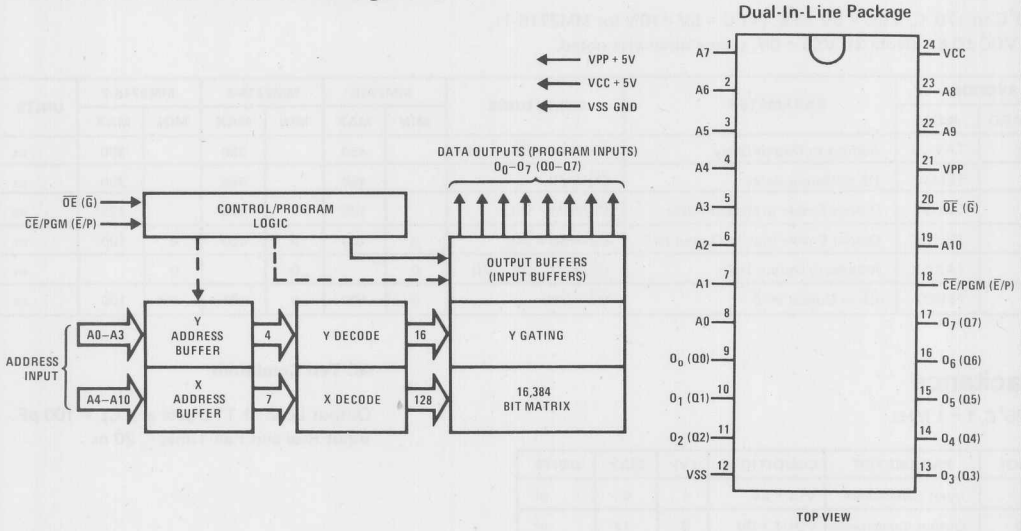
The MM2716 is packaged in a 24-pin dual-in-line package with transparent lid. The transparent lid allows the user to expose the chip to ultraviolet light to erase the bit pattern. A new pattern can then be written into the device by following the programming procedure.

This EPROM is fabricated with the reliable, high volume, time proven, N-channel silicon gate technology.

Features

- 2048 x 8 organization
- 525 mW max active power, 132 mW max standby power
- Low power during programming
- Access time—MM2716, 450 ns; MM2716-1, 350 ns; MM2716-2, 390 ns
- Single 5V power supply
- Static—no clocks required
- Inputs and outputs TTL compatible during both read and program modes
- TRI-STATE® output

Block and Connection Diagrams *



Pin Connection During Read or Program

MODE	PIN NAME/NUMBER				
	CE/PGM (E/P) 18	OE (G) 20	VPP 21	VCC 24	OUTPUTS 9–11, 13–17
Read	VIL	VIL	5	5	DOUT
Program	Pulsed VIL to VIH	VIH	25	5	DIN

Pin Names

A0–A10	Address Inputs
O0–O7 (Q0–Q7)	Data Outputs
CE/PGM (E/P)	Chip Enable/Program
OE (G)	Output Enable
VPP	Read 5V, Program 25V
VCC	Power (5V)
VSS	Ground

*Symbols in parentheses are proposed JEDEC standard.

Absolute Maximum Ratings (Note 1)

Temperature Under Bias	−25°C to +85°C	All Input or Output Voltages with Respect to VSS (except VPP)	6V to −0.3V
Storage Temperature	−65°C to +125°C	Power Dissipation	1.5 W
VPP Supply Voltage with Respect to VSS	26.5V to −0.3V	Lead Temperature (Soldering, 10 seconds)	300°C

READ OPERATION

DC Operating Characteristics

T_A = 0°C to +70°C, VCC = 5V ±5%, (VCC = 5V ±10% for MM2716-1),
VPP = VCC ±0.6V (Note 3), VSS = 0V, unless otherwise noted.

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
ILI	Input Current	VIN = 5.25V or VIN = VIL			10	μA
ILO	Output Leakage Current	VOUT = 5.25V, $\overline{CE}$/PGM = 5V			10	μA
IPP1	VPP Supply Current	VPP = 5.85V			5	mA
ICC1	VCC Supply Current (Standby)	$\overline{CE}$/PGM = VIH, $\overline{OE}$ = VIL		10	25	mA
ICC2	VCC Supply Current (Active)	$\overline{CE}$/PGM = $\overline{OE}$ = VIL		57	100	mA
VIL	Input Low Voltage		0.1		0.8	V
VIH	Input High Voltage		2.0		V_{CC} + 1	V
VOH	Output High Voltage	IOH = 400 μA	2.4			V
VOL	Output Low Voltage	IOL = 2.1 mA			0.45	V

AC Characteristics (Note 2)

T_A = 0°C to +70°C, VCC = 5V ±5%, (VCC = 5V ±10% for MM2716-1),
VPP = VCC ±0.6V (Note 3), VSS = 0V, unless otherwise noted.

SYMBOL		PARAMETER	CONDITIONS	MM2716		MM2716-1		MM2716-2		UNITS
STANDARD	JEDEC			MIN	MAX	MIN	MAX	MIN	MAX	
t_{ACC}	TAVQV	Address to Output Delay	$\overline{CE}$/PGM = $\overline{OE}$ = VIL		450		350		390	ns
t_{CE}	TELQV	$\overline{CE}$ to Output Delay	$\overline{OE}$ = VIL		450		350		390	ns
t_{OE}	TGLQV	Output Enable to Output Delay	$\overline{CE}$/PGM = VIL		120		120		120	ns
t_{DF}	TGHQZ	Output Enable High to Output Hi-Z	$\overline{CE}$/PGM = VIL	0	100	0	100	0	100	ns
t_{OH}	TAXQX	Address to Output Hold	$\overline{CE}$/PGM = $\overline{OE}$ = VIL	0		0		0		ns
t_{OD}	TEHQZ	$\overline{CE}$ to Output Hi-Z	$\overline{OE}$ = VIL	0	100	0	100	0	100	ns

Capacitance

T_A = 25°C, f = 1 MHz

SYMBOL	PARAMETER	CONDITIONS	TYP	MAX	UNITS
CI	Input Capacitance	VIN = 0V	4	6	pF
CO	Output Capacitance	VOUT = 0V	8	12	pF

AC Test Conditions

Output Load: 1 TTL gate and CL = 100 pF.
Input Rise and Fall Times ≤ 20 ns

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. Except for "Operating Temperature Range" they are not meant to imply that the devices should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

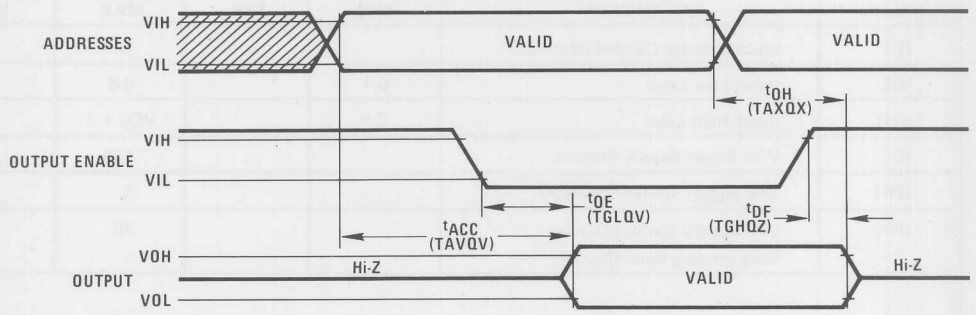
Note 2: Typical conditions are for operation at: T_A = 25°C, VCC = 5V, VPP = VCC, and VSS = 0V.

Note 3: VPP may be connected to VCC except during program. The ±0.6V tolerance allows a circuit to switch VPP between the read voltage and the program voltage.

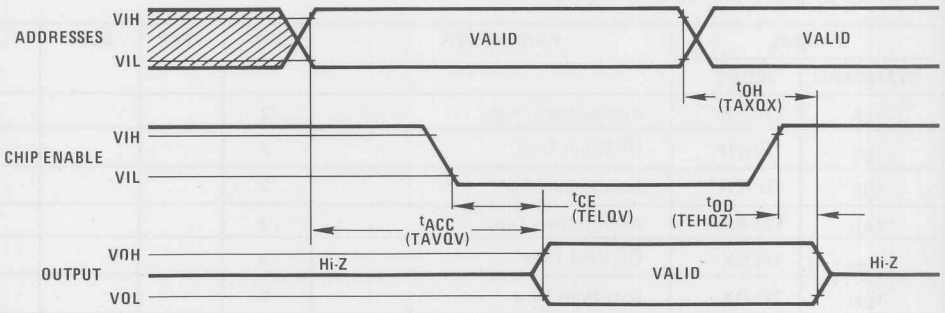
Note 4: Capacitance is guaranteed by periodic testing. T_A = 25°C, f = 1 MHz.

Switching Time Waveforms*

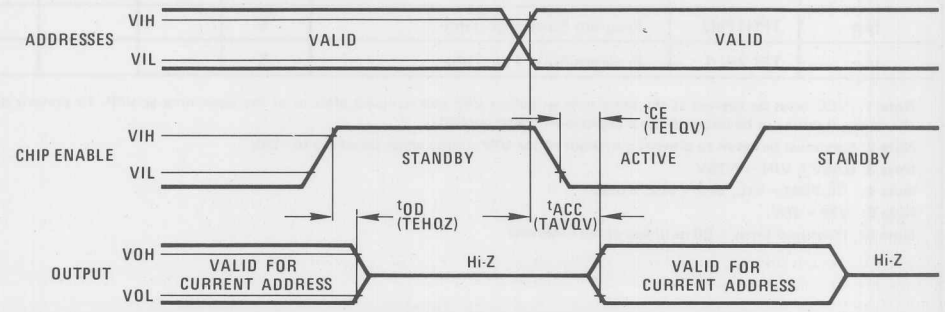
Read Cycle ($\overline{CE}$/PGM = VIL)



Read Cycle ($\overline{OE}$ = VIL)



Standby Power Down Mode ($\overline{OE}$ = VIL)



*Symbols in parentheses are proposed JEDEC standard.

PROGRAM OPERATION

DC Electrical Characteristics and Operating Conditions (Notes 1 and 2)

(T_A = 25°C ±5°C) (VCC = 5V ±5%, VPP = 25V ±1V)

SYMBOL	PARAMETER	MIN	TYP	MAX	UNITS
ILI	Input Leakage Current (Note 3)			10	μA
VIL	Input Low Level	−0.1		0.8	V
VIH	Input High Level	2.0		VCC + 1	V
ICC	VCC Power Supply Current			100	mA
IPP1	VPP Supply Current (Note 4)			5	mA
IPP2	VPP Supply Current During Programming Pulse (Note 5)			30	mA

AC Characteristics and Operating Conditions (Notes 1, 2, and 6)

(T_A = 25°C ±5°C) (VCC = 5V ±5%, VPP = 25V ±1V)

SYMBOL		PARAMETER	MIN	TYP	MAX	UNITS
STANDARD	JEDEC					
t_{AS}	TAVPH	Address Setup Time	2			μs
t_{OS}	TGHPH	$\overline{OE}$ Setup Time	2			μs
t_{DS}	TDVPH	Data Setup Time	2			μs
t_{AH}	TPLAX	Address Hold Time	2			μs
t_{OH}	TPLGX	$\overline{OE}$ Hold Time	2			μs
t_{DH}	TPLDX	Data Hold Time	2			μs
t_{DF}	TGHQZ	Chip Disable to Output Float Delay (Note 4)	0		100	ns
t_{CE}	TGLQV	Chip Enable to Output Delay (Note 4)			120	ns
t_{PW}	TPHPL	Program Pulse Width	45	50	55	ms
t_{PR}	TPH1PH2	Program Pulse Rise Time	5			ns
t_{PF}	TPL2PL1	Program Pulse Fall Time	5			ns

Note 1: VCC must be applied at the same time or before VPP and removed after or at the same time as VPP. To prevent damage to the device it must not be inserted into a board with power applied.

Note 2: Care must be taken to prevent overshoot of the VPP supply when switching to +25V.

Note 3: 0.45V ≤ VIN ≤ 5.25V.

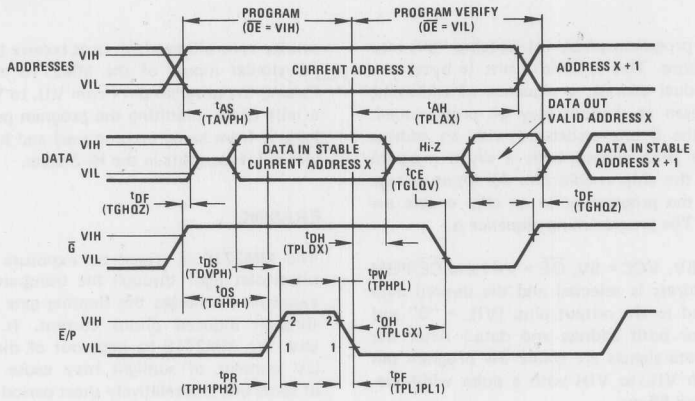
Note 4: $\overline{CE}$/PGM = VIL, VPP = VCC + 0.6V.

Note 5: VPP = 26V.

Note 6: Transition times ≤ 20 ns unless noted otherwise.

Timing Diagram*

Program Mode



DEVICE OPERATION

The MM2716 has 3 modes of operation in the normal system environment. These are shown in Table I.

Read Mode

The MM2716 read operation requires that $\overline{OE}$ = VIL, $\overline{CE}$/PGM = VIL and that addresses A0–A10 have been stabilized. Valid data will appear on the output pins after t_{ACC}, t_{OE} or t_{CE} times (see Switching Time Waveforms) depending on which is limiting.

Deselect Mode

The MM2716 is deselected by making $\overline{OE}$ = VIH. This mode is independent of $\overline{CE}$/PGM and the condition of the addresses. The outputs are Hi-Z when $\overline{OE}$ = VIH. This allows OR-tying 2 or more MM2716's for memory expansion.

Standby Mode (Power Down)

The MM2716 may be powered down to the standby mode by making $\overline{CE}$/PGM = VIH. This is independent of $\overline{OE}$ and automatically puts the outputs in their Hi-Z state. The power is reduced to 25% (132 mW max) of the normal operating power. VCC and VPP must be maintained at 5V. Access time at power up remains either t_{ACC} or t_{CE} (see Switching Time Waveforms).

PROGRAMMING

The MM2716 is shipped from National completely erased. All bits will be at a "1" level (output high) in this initial state and after any full erasure. Table II shows the 3 programming modes.

TABLE I. OPERATING MODES (VCC = VPP = 5V)

MODE	PIN NAME/NUMBER		
	$\overline{CE}$/PGM ($\overline{E}$/P) 18	$\overline{OE}$ ($\overline{G}$) 20	OUTPUTS 9–11, 13–17
Read	VIL	VIL	DOUT
Deselect	Don't Care	VIH	Hi-Z
Standby	VIH	Don't Care	Hi-Z

TABLE II. PROGRAMMING MODES (VCC = 5V)

MODE	PIN NAME/NUMBER			
	$\overline{CE}$/PGM ($\overline{E}$/P) 18	$\overline{OE}$ ($\overline{G}$) 20	VPP 21	OUTPUTS O 9–11, 13–17
Program	Pulsed VIL to VIH	VIH	25	DIN
Program Verify	VIL	VIL	25(5)	DOUT
Program Inhibit	VIL	VIH	25	Hi-Z

*Symbols in parentheses are proposed JEDEC standard

Program Mode

The MM2716 is programmed by introducing "0"s into the desired locations. This is done 8 bits (a byte) at a time. Any individual address, a sequence of addresses, or addresses chosen at random may be programmed. Any or all of the 8 bits associated with an address location may be programmed with a single program pulse applied to the chip enable pin. All input voltage levels, including the program pulse on chip enable are TTL compatible. The programming sequence is:

With VPP = 25V, VCC = 5V, $\overline{OE}$ = VIH and $\overline{CE}$/PGM = VIL, an address is selected and the desired data word is applied to the output pins. (VIL = "0" and VIL = "1" for both address and data.) After the address and data signals are stable the program pin is pulsed from VIL to VIH with a pulse width between 45 ms and 55 ms.

Multiple pulses are not needed but will not cause device damage. No pins should be left open. A high level (VIH or higher) *must not* be maintained longer than $t_{PW(MAX)}$ on the program pin during programming. MM2716's may be programmed in parallel with the same data in this mode.

Program Verify Mode

The programming of the MM2716 may be verified either 1 word at a time during the programming (as shown in the timing diagram) or by reading all of the words out at the end of the programming sequence. This can be done with VPP = 25V (or 5V) in either case. VPP must be at 5V for all operating modes and can be maintained at 25V for all programming modes.

Program Inhibit Mode

The program inhibit mode allows programming several MM2716's simultaneously with different data for each one by controlling which ones receive the program pulse. All similar inputs of the MM2716 may be paralleled. Pulsing the program pin (from VIL to VIH) will program a unit while inhibiting the program pulse to a unit will keep it from being programmed and keeping $\overline{OE}$ = VIH will put its outputs in the Hi-Z state.

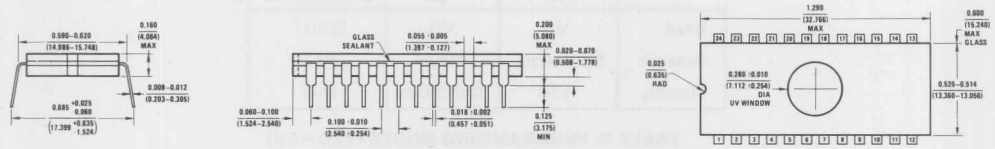
ERASING

The MM2716 is erased by exposure to high intensity ultraviolet light through the transparent window. This exposure discharges the floating gate to its initial state through induced photo current. It is recommended that the MM2716 be kept out of direct sunlight. The UV content of sunlight may cause a partial erasure of some bits in a relatively short period of time.

An ultraviolet source of 2537 Å yielding a total integrated dosage of 15 watt-seconds/cm^2 is required. This will erase the part in approximately 15 to 20 minutes if a UV lamp with a 12,000 μW/cm^2 power rating is used. The MM2716 to be erased should be placed 1 inch away from the lamp and no filters should be used.

An erasure system should be calibrated periodically. The distance from lamp to unit should be maintained at 1 inch. The erasure time is increased by the square of the distance (if the distance is doubled the erasure time goes up by a factor of 4). Lamps lose intensity as they age. When a lamp is changed, the distance is changed, or the lamp is aged, the system should be checked to make certain full erasure is occurring. Incomplete erasure will cause symptoms that can be misleading. Programmers, components, and system designs have been erroneously suspected when incomplete erasure was the basic problem.

Physical Dimensions inches (millimeters)



UV Window Cavity Dual-In-Line Package (JQ)
Order Number MM2716Q
NS Package Number J24CQ

National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, California 95051
Tel.: (408) 737-5000
TWX: (910) 339-9240

National Semiconductor GmbH
Eisenheimerstrasse 61/II
8000 München 21
West Germany
Tel.: (089) 576091
Telex: 05-22772

NS International Inc., Japan
Miyake Building
1-9 Yotsuya, Shinjuku-ku 160
Tokyo, Japan
Tel.: (03) 355-3711
TWX: 232-2015 NSCJ-J

National Semiconductor (Hong Kong) Ltd.
8th Floor,
Cheung Kong Electronic Bldg.
4 Hing Yip Street
Kwun Tong
Kowloon, Hong Kong
Tel.: 3-899235
Telex: 73866 NSEHK HX
Cable: NATSEMI

NS Electronics Do Brasil
Avda Brigadeiro Faria Lima 844
11 Andar Conjunto 1104
Jardim Paulistano
Sao Paulo, Brasil
Telex:
1121008 CABINE SAO PAULO

NS Electronics Pty. Ltd.
Cnr. Stud Rd. & Mtn. Highway
Bayswater, Victoria 3153
Australia
Tel.: 03-729-6333
Telex: 32096

National Semiconductor

MARCH 1979
PRELIMINARY

MM2708M 8k UV Erasable PROM Military Temperature Range

General Description

The MM2708M is a high speed 8192-Bit UV erasable and electrically reprogrammable EPROM ideally suited for applications where fast turn-around and pattern experimentation are important requirements.

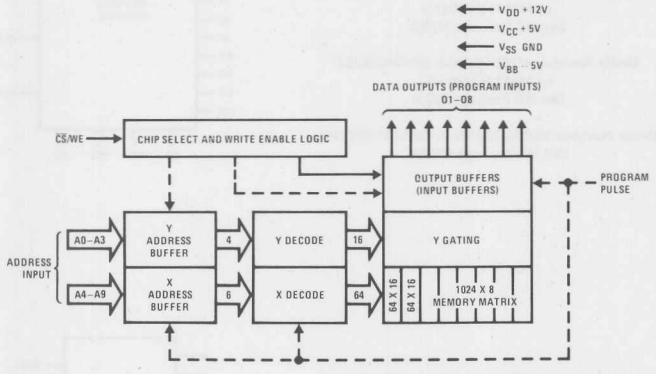
The MM2708M is packaged in a 24-pin dual-in-line package with transparent lid. The transparent lid allows the user to expose the chip to ultraviolet light to erase the bit pattern. A new pattern can then be written into the device by following the programming procedure.

The MM2708M is fabricated with the reliable, high volume, time proven, N-channel silicon gate technology.

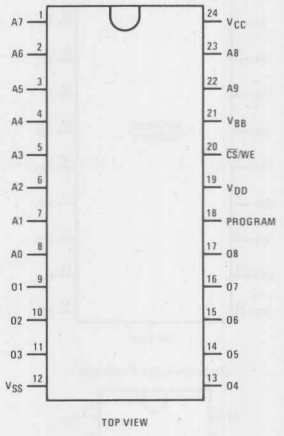
Features

- −55°C to +125°C operation
- 1024 x 8 organization
- 800 mW max
- Low power during programming
- Access time 450 ns
- Standard power supplies: 12V, 5V, −5V
- Static—no clocks required
- Inputs and outputs TTL compatible during both read and program modes
- TRI-STATE® output

Block and Connection Diagrams



Dual-In-Line Package



Order Number MM2708MQ

Pin Connection During Read or Program

MODE	PIN NUMBER						
	9–11, 13–17	12	18	19	20	21	24
Read	D_{OUT}	V_{SS}	V_{SS}	V_{DD}	V_{IL}	V_{BB}	V_{CC}
Program	D_{IN}	V_{SS}	Pulsed V_{IHP}	V_{DD}	V_{IHW}	V_{BB}	V_{CC}

Pin Description

A0–A9	Address inputs
O1–O8	Data outputs
CS/WE	Chip select/write enable input

National Semiconductor

CMOS RAMs

MM54C920/MM74C920 1024-Bit (256 x 4) Static RAM
MM54C921/MM74C921 1024-Bit (256 x 4) Static RAM

General Description

The MM54C920/MM74C920 256 x 4 random access read/write memory is manufactured using silicon gate CMOS technology. Data output is the same polarity as data input. Internal latches store address inputs, $\overline{CES}$ and data output. This RAM is specifically designed to operate from standard 54/74 TTL power supplies. All inputs and outputs are TTL compatible.

The MM54C921/MM74C921 is identical to the MM54C920/MM74C920, except data inputs are internally connected to data outputs; the number of package leads thereby is reduced to 18.

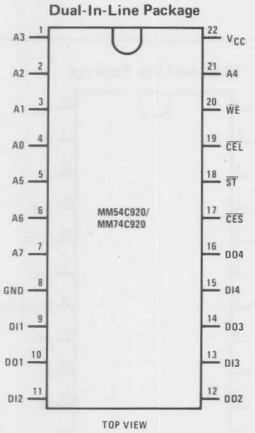
Complete address decoding as well as 2-chip select functions, $\overline{CEL}$ and $\overline{CES}$, and TRI-STATE® outputs allow easy expansion with a minimum of external components. Versatility plus high speed and low power make these RAMs ideal elements for use in microprocessor, minicomputer as well as main frame memory applications.

Features

- 256 x 4-bit organization
- Access time
  - 250 ns max MM74C920, MM74C921
  - 275 ns max MM54C920, MM54C921
  - 300 ns max MM74C921-3
- TRI-STATE outputs
- Low power
- On-chip registers
- Single 5V supply
- Data retained with V_{CC} as low as 2V

Connection Diagrams

Dual-In-Line Package



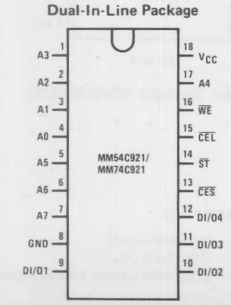
TOP VIEW

Order Number MM54C920D, MM74C920D
or MM74C920D-3
See NS Package D22B

Order Number MM54C920J, MM74C920J
or MM74C920J-3
See NS Package J22A

Order Number MM74C920N or MM74C920N-3
See NS Package N22A

Dual-In-Line Package



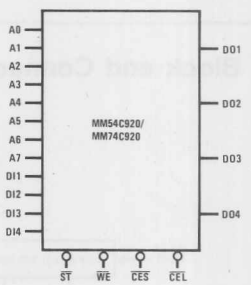
TOP VIEW

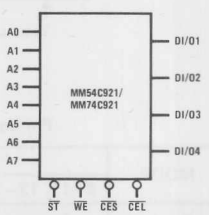
Order Number MM54C921D, MM74C921D
or MM74C921D-3
See NS Package D18A

Order Number MM54C921J, MM74C921J
or MM74C921J-3
See NS Package J18A

Order Number MM74C921N or MM74C921N-3
See NS Package N18A

Logic Symbols





National Semiconductor

November 1979

MM52116 (MM2316E) 16,384-Bit Read Only Memory

General Description

The MM52116 is a static MOS 16,384-bit read-only memory organized in an 2048-word-by-8-bit format. It is fabricated using N-channel enhancement and depletion-mode technology which provides complete DTL/TTL compatibility and single power-supply operation.

Three programmable chip selects controlling the TRI-STATE® outputs allow for memory expansion.

Programming of the memory array and chip-select active levels is accomplished by changing two masks during fabrication.

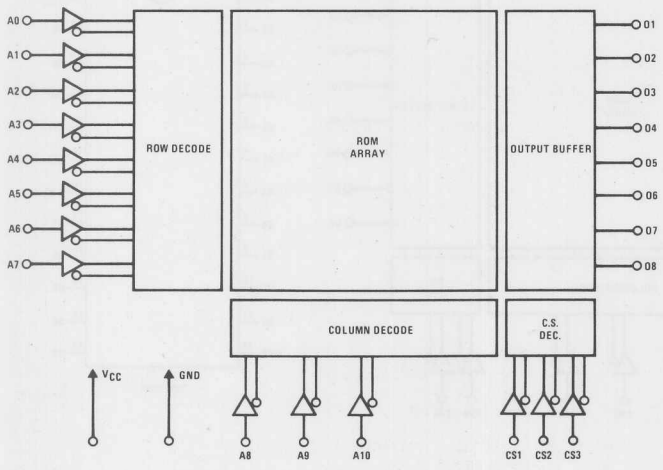
Features

- Fully decoded
- Single 5V power supply ±10% tolerance
- Inputs and outputs TTL compatible
- Outputs drive 2 TTL loads and 100 pF
- Static operation
- TRI-STATE outputs for bus interface
- Programmable chip selects
- 2048-word-by-8-bit organization
- Maximum access time — 450 ns
- Industry standard pin outs (MM2316E)

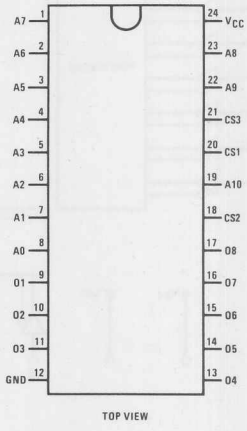
Applications

- Microprocessor instruction store
- Control logic
- Table look-up

Block and Connection Diagrams



Dual-In-Line Package



TOP VIEW

National Semiconductor

November 1979

MM52132 32,768-Bit (4096 × 8) MAXI-ROM™

General Description

The MM52132 is a static MOS 32,768-bit read-only memory organized in a 4096-word-by-8-bit format. It is fabricated using N-channel enhancement and depletion-mode technology which provides complete DTL/TTL compatibility and single power-supply operation.

Two programmable chip selects controlling the TRI-STATE® outputs allow for memory expansion.

Programming of the memory array and chip-select active levels is accomplished by changing two masks during fabrication.

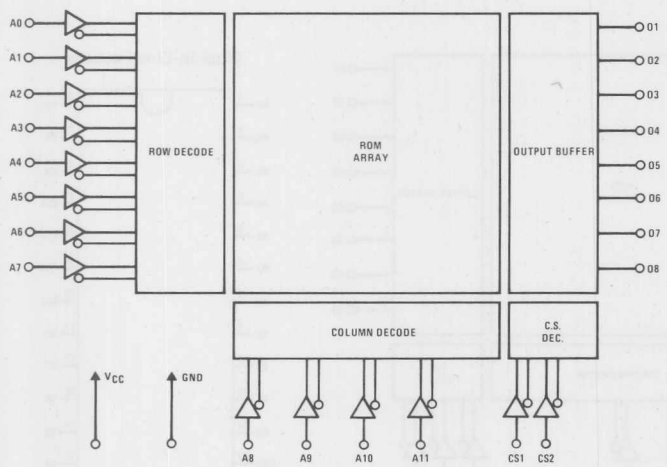
Features

- Fully decoded
- Single 5V power supply ±10% tolerance
- Inputs and outputs TTL compatible
- Outputs drive 2 TTL loads and 100 pF
- Static operation
- TRI-STATE outputs for bus interface
- Programmable chip selects
- 4096-word-by-8-bit organization
- Maximum access time — 450 ns
- Industry standard pin outs

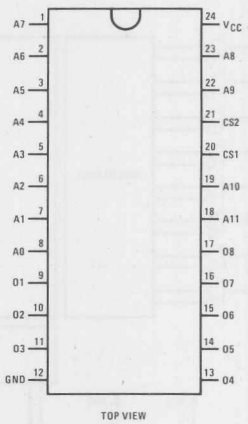
Applications

- Microprocessor instruction store
- Control logic
- Table look-up

Block and Connection Diagrams



Dual-In-Line Package



TOP VIEW

National Semiconductor

November 1979

MM52164 65,536-Bit (8192 × 8) MAXI-ROM™

General Description

The MM52164 is a static MOS 65,536-bit read-only memory organized in an 8192-word by 8-bit format. It is fabricated using N-channel enhancement and depletion-mode technology which provides complete DTL/TTL compatibility and single power-supply operation.

One programmable chip select controlling the TRI-STATE® outputs allow for memory expansions.

Programming of the memory array and chip-select active levels is accomplished by changing two masks during fabrication.

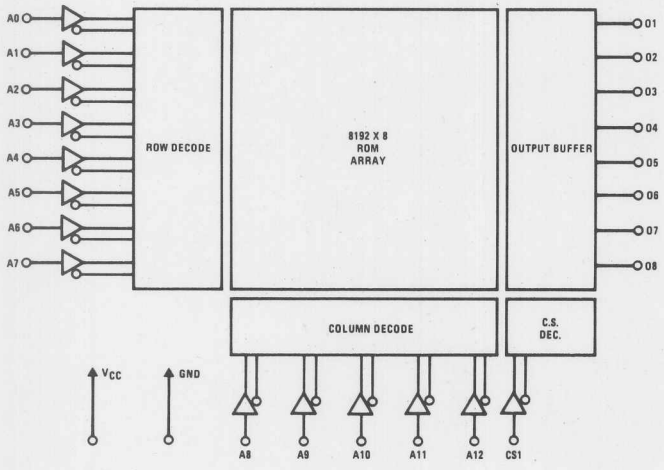
Features

- Fully decoded
- Single 5V power supply ±10% tolerance
- Inputs and outputs TTL compatible
- Outputs drive 2 TTL loads and 100 pF
- Static operation
- TRI-STATE outputs for bus interface
- Programmable chip select
- 8192-word-by-8-bit organization
- Maximum access time — 450 ns
- Industry standard pin outs

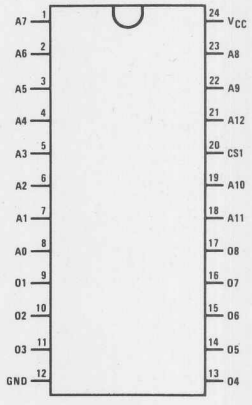
Applications

- Microprocessor instruction store
- Control logic
- Table look-up

Block and Connection Diagrams



Dual-In-Line Package



TOP VIEW

Peripheral Control Components

National Semiconductor

Preliminary

JULY 1978

INS8253 Programmable Interval Timer

General Description

The INS8253 is a programmable timer/counter device contained in a standard, 24-pin dual-in-line package. The chip, which is fabricated using N-channel silicon gate technology, provides counting or time-out services in a microcomputer system. The various operating modes and other functional characteristics of the INS8253 are programmed by the system software.

The INS8253 provides three independent 16-bit down counters, each of which is capable of count rates in the range DC to 2MHz. Through software initialization, each counter can be made to operate in any one of six modes. The modulus and counting system used are also specified by system software. The operating characteristics of any individual counter can be modified by the software at any time to meet changing system requirements.

The modulus of any given counter can be changed at the program's discretion by loading a new value into the counter. A counter load operation may be limited to the counter's least significant byte or to its most significant byte, or it may revise both halves of the counter.

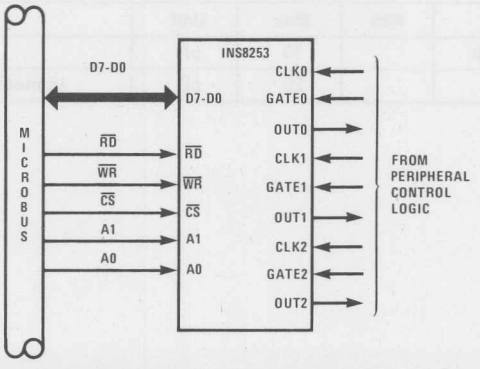
Count sequences may be in either binary or BCD. This choice is also individually specified for each counter by the software.

The contents of each counter may be read either directly or through an auxiliary register. A direct reading of the counter can be made whenever the counter is inhibited from counting. A count value can also be read without interfering with the counting process. This is done by transferring the counter's current value to an auxiliary register and then reading that register. This counter-to-register transfer can be executed without affecting the normal count sequence.

Features

- 3 Individually Programmable 16-Bit Counters
- 6 Operating Modes
- DC to 2MHz Count Rates
- Individual Count Rate and Modulus for Each Counter
- Selectable Counting System (Binary or BCD) for Each Counter
- TRI-STATE® TTL Drive Capability for Bidirectional Data Bus
- Single +5 Volt Power Supply
- 24-Pin Dual-in-Line Package
- MICROBUS™* Compatible

INS8253 MICROBUS Configuration



*A trademark of National Semiconductor Corporation.

Absolute Maximum Ratings

Ambient Temperature Under Bias ..... 0°C to +70°C
Maximum Voltage to Any Input
with Respect to GND ..... −0.5V to +7V
Storage Temperature ..... −65°C to +150°C
Power Dissipation ..... 1 Watt

Note: Absolute maximum ratings indicate limits beyond which permanent damage may occur. Continuous operation at these limits is not intended and should be limited to those conditions specified under DC Electrical Characteristics.

DC Electrical Characteristics

(T_A = 0°C to +70°C; V_{CC} = 5V ± 5%)

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.2	V_{CC} + 0.5V	V	
V_{OL}	Output Low Voltage		0.45	V	Note 1
V_{OH}	Output High Voltage	2.4		V	Note 2
I_{IL}	Input Load Current		±10	μA	V_{IN} = V_{CC} to 0V
I_{OFL}	Output Float Leakage		±10	μA	V_{OUT} = V_{CC} to 0V
I_{CC}	V_{CC} Supply Current		140	mA	

Note 1: INS8253, I_{OL} = 1.6mA.
Note 2: INS8253, I_{OH} = −150μA.

Capacitance

T_A = 25°C; V_{CC} = GND = 0V.

Symbol	Parameter	Min	Max	Unit	Test Conditions
C_{IN}	Input Capacitance		10	pF	f_C = 1MHz
$C_{I/O}$	I/O Capacitance		20	pF	Unmeasured pins returned to V_{SS}

AC Electrical Characteristics

T_A = 0°C to +70°C; V_{CC} = 5.0V ± 5%; GND = 0V

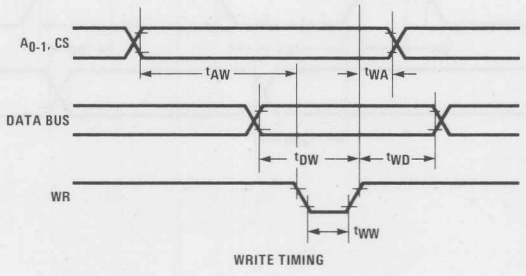
Bus Parameters:

Symbol	Parameter	Min	Max	Unit
READ CYCLE				
t_{AR}	Address Stable Before $\overline{READ}$	50		ns
t_{RA}	Address Hold Time for $\overline{READ}$	5		ns
t_{RR}	$\overline{READ}$ Pulse Width	400		ns
t_{RD}	Data Delay from $\overline{READ}$ (Note 2)		300	ns
t_{DF}	$\overline{READ}$ to Data Floating	25	125	ns
WRITE CYCLE				
t_{AW}	Address Stable Before $\overline{WRITE}$	50		ns
t_{WA}	Address Hold Time for $\overline{WRITE}$	30		ns
t_{WW}	$\overline{WRITE}$ Pulse Width	400		ns
t_{DW}	Data Setup Time for $\overline{WRITE}$	300		ns
t_{WD}	Data Hold Time for $\overline{WRITE}$	40		ns
t_{RV}	Recovery Time Between $\overline{WRITE}$s	1		ns

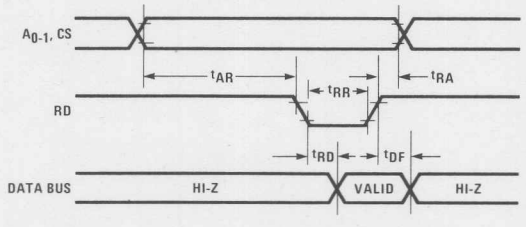
Note 1: AC timings measured at V_{OH} = 2.2V, V_{OL} = 0.8V.

Note 2: Test conditions: INS8253, C_L = 100pF.

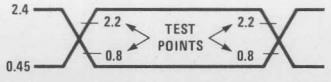
Input Waveforms for AC Tests



WRITE TIMING



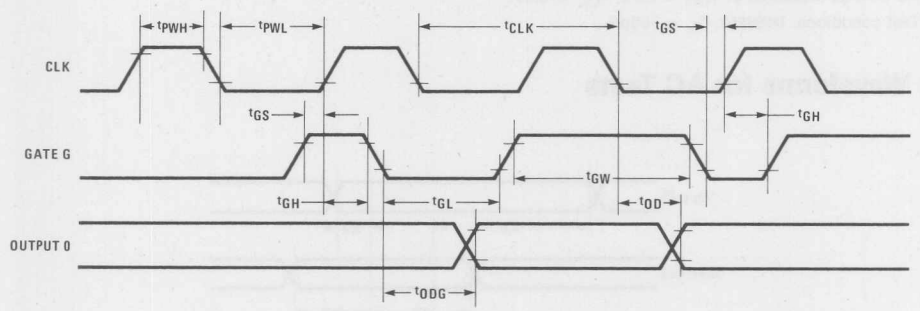
READ TIMING



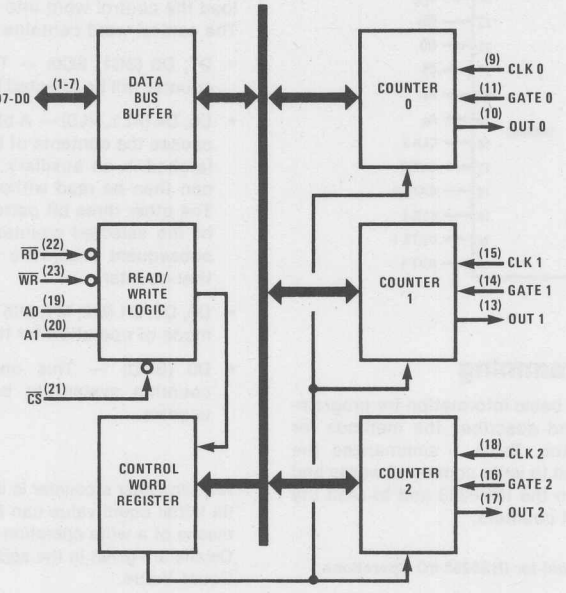
Clock and Gate Timing

Symbol	Parameter	Min	Max	Unit
t_{CLK}	Clock Period	380	DC	ns
t_{PWH}	High Pulse Width	230		ns
t_{PWL}	Low Pulse Width	150		ns
t_{GW}	Gate Width High	150		ns
t_{GL}	Gate Width Low	100		ns
t_{GS}	Gate Setup Time to CLK ↑	100		ns
t_{GH}	Gate Hold Time After CLK ↑	50		ns
t_{OD}	Output Delay From CLK ↓ (Note 1)		400	ns
t_{ODG}	Output Delay From Gate ↓ (Note 1)		300	ns

Note 1: Test conditions: INS8253, C_L = 100pF.



INS8253 Functional Block Diagram



NOTE: APPLICABLE PINOUT NUMBERS ARE INCLUDED WITHIN PARENTHESES.

INS8253 Functional Pin Description

The following describes the functions of all INS8253 input/output pins. Some of these descriptions refer to internal circuits.

NOTE

In the following descriptions, a low represents a logic 0 (0 Volt, nominal) and a high represents a logic 1 (+2.4 Volts, nominal).

INPUT SIGNALS

Chip Select ($\overline{CS}$): When low, the chip is selected. This enables communication between the INS8253 and the microprocessor.

Read ($\overline{RD}$): When low, allows the microprocessor to read contents of counter specified by A0, A1.

Write ($\overline{WR}$): When low, writes control word into control word register or loads new count value into selected counter. Destination of data (control word register or counter 0, 1 or 2) is specified by A0, A1.

A0, A1: These inputs are used to select one of the counters for reading or writing or to select the control word register for writing. A0, A1 may be controlled via address bus lines.

Clock (CLK0-CLK2): Each counter has a separate clock input that drives the counter.

Gate (Gate 0-Gate 2): Each counter is individually controlled by a separate Gate input (1 = enable, 0 = inhibit). In some modes, the positive edge of Gate is used to initiate the counting process. Specific use of Gate depends on the counter's operating mode. Details are provided in the section entitled INS8253 Programming.

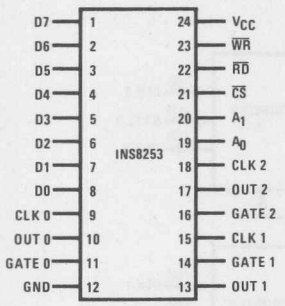
OUTPUT SIGNALS

Output (Out 0-Out 2): Each counter has a single output that indicates whether or not the counter has reached its terminal count. Specific operation of this output depends on the counter's mode. Details are provided in the section entitled INS8253 Programming.

INPUT/OUTPUT SIGNALS

Data (D7-D0) Bus: This bus, which comprises eight TRI-STATE input/output lines, provides for bidirectional communication between the INS8253 and the microprocessor. Control words and count value bytes are transferred over these lines.

INS8253 Pin Configuration



INS8253 Programming

This section provides basic information for programming the INS8253 and describes the methods for reading counter status. Table 1 summarizes the control signals needed to write command words and new count values into the INS8253 and to read the contents of individual counters.

Table 1. Bus Control for INS8253 I/O Operations

Output Operations	$\overline{CS}$	$\overline{WR}$	$\overline{RD}$	A1	A0
LOAD COUNTER 0	0	0	1	0	0
LOAD COUNTER 1	0	0	1	0	1
LOAD COUNTER 2	0	0	1	1	0
WRITE CONTROL WORD	0	0	1	1	1
Input Operations					
READ COUNTER 0	0	1	0	0	0
READ COUNTER 1	0	1	0	0	1
READ COUNTER 2	0	1	0	1	0

WRITING CONTROL WORDS

Each counter's mode and counting system (binary or BCD) are specified by an eight-bit control word. See figure 1. An I/O write operation with A0, A1 = 11 will load the control word into the control word register. The control word contains four fields:

- D7, D6 (SC1, SC0) — This field specifies which counter will be affected by the other control fields.
- D5, D4 (RL1, RL0) — A bit pattern of 00 in this field causes the contents of the selected counter to be latched in an auxiliary register. The count value can then be read without inhibiting the counter. The other three bit patterns specify which byte(s) of the selected counter will be affected by any subsequent read/write operations addressed to that counter.
- D3, D2, D1 (M2, M1, M0) — This field specifies the mode of operation for the selected counter.
- D0 (BCD) — This one-bit field specifies the counting system to be used by the selected counter.

Any time after a counter is initialized by a control word, its initial count value can be loaded. This is done by means of a write operation addressed to that counter. Details are given in the section entitled Loading Initial Count Value.

Programming of the three counters can be executed in any sequence, with only two requirements.

1. A counter must be issued a control word before it is given an initial count value.
2. Read and write operations addressed to a counter must conform to the byte-selection rules specified by the RL1, RL0 field in the control word. For example, if the counter's RL1, RL0 bits = 10, subsequent counter load operations addressed to that counter must be intended for the most significant byte only.

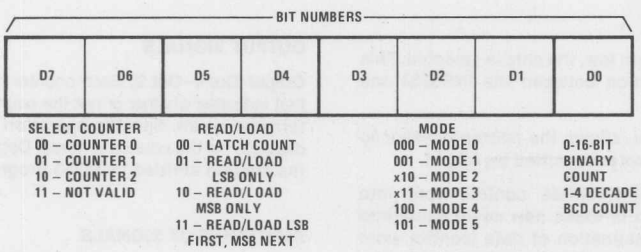


Figure 1. Control Word Format

COUNTER MODE DESCRIPTIONS

Figure 2 provides timing information for the six INS8253 operating modes.

- Mode 0, Timed Interrupt — In this mode, OUT goes low when the mode is set. The counter begins counting CLK cycles when the count is loaded. OUT remains low until the terminal count is reached, at which point it goes high and remains high until either the mode or count is reloaded. The gate input will inhibit the count when low.

  If the counter is loaded with a new value during a count cycle, counting will stop when the first byte is loaded and will begin decrementing from the new value after the second byte is loaded.

- Mode 1, Retriggerable One Shot — In this mode, OUT goes low on the first CLK after a rising transition on Gate. OUT goes high again on the terminal count. Gate can be used to retrigger the counter. Each positive transition of Gate causes the counter to begin decrementing from the initial count value.

  If a new initial count value is loaded during a count cycle, the new value will not take effect until the next rising transition of Gate.

- Mode 2, Rate Generator — In this mode, OUT goes low for one CLK cycle at the end of each count sequence. The leading edge of each pulse occurs at the start of the terminal CLK cycle. The counter will repeat count sequences as long as Gate remains high. Any positive transition of Gate will start a new count sequence at the initial count value. This allows the counter to be synchronized by Gate.

  If a new initial count value is loaded during a count sequence, the current sequence will run to completion and the following sequence will then start at the new initial count value.

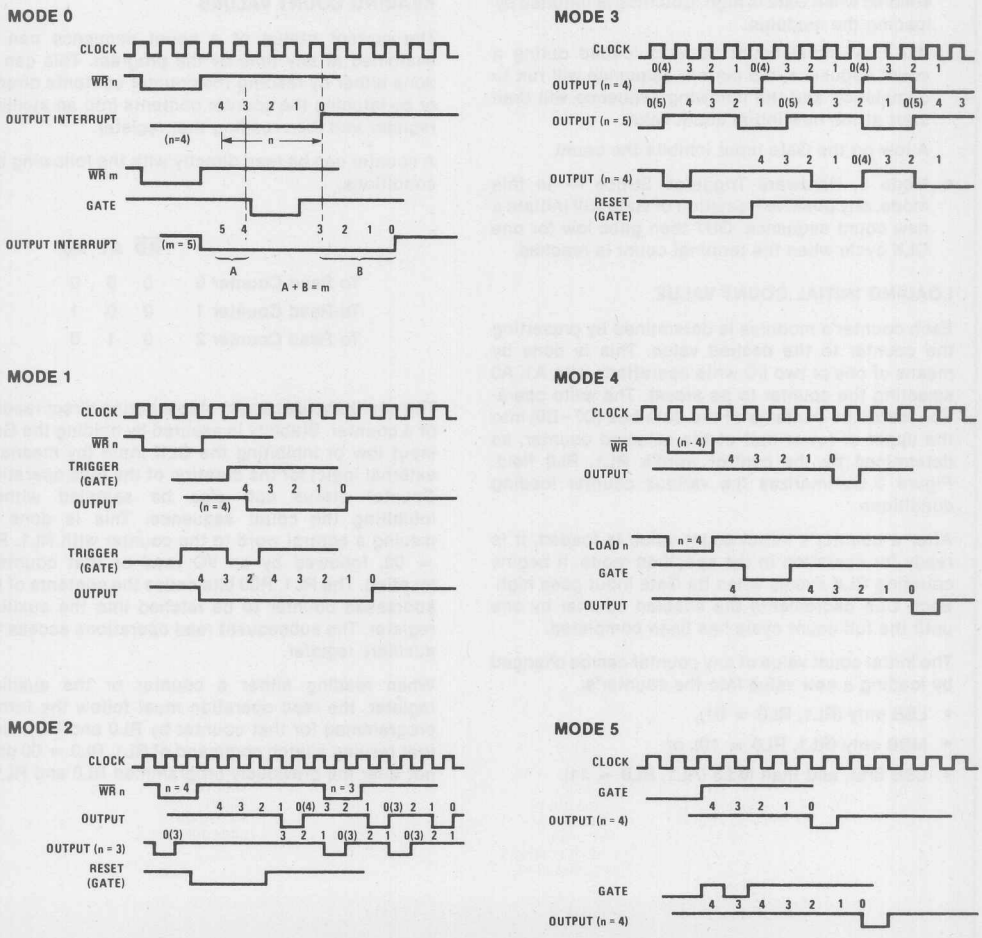


Figure 2. Mode Timing Waveforms

- Mode 3, Square Wave Generator — In this mode, the counter generates a square wave signal at the OUT pin so long as Gate remains high. The period of the square wave is equal to one count cycle. If the initial count value is even, OUT will be high for the first half of each count sequence and low during each second half. For an odd count, OUT is high for (N + 1)/2 counts and low for (N – 1)/2 counts.

  If a new initial count value is loaded during a count sequence, the current sequence will run to completion and the following sequence will then start at the new initial count value.

  Any positive transition of Gate will start a new count sequence at the initial count value. This allows the counter to be synchronized by Gate.

- Mode 4, Software Triggered Strobe — In this mode, OUT is normally high and goes low for one CLK cycle after the terminal count is reached. Counting is enabled when Gate is high. Counting is initiated by loading the modulus.

  If a new initial count value is loaded during a count sequence, the current sequence will run to completion and the following sequence will then start at the new initial count value.

  A low on the Gate input inhibits the count.

- Mode 5, Hardware Triggered Strobe — In this mode, any positive transition of Gate will initiate a new count sequence. OUT then goes low for one CLK cycle when the terminal count is reached.

LOADING INITIAL COUNT VALUE

Each counter's modulus is determined by presetting the counter to the desired value. This is done by means of one or two I/O write operations with A1, A0 selecting the counter to be preset. The write operation loads the contents of the data bus (D7-D0) into the upper or lower half of the selected counter, as determined by the control word's RL1, RL0 field. Figure 3 summarizes the various counter loading conditions.

After a counter's initial count value is loaded, it is ready for operation in the specified mode. It begins counting CLK cycles when its Gate input goes high. Each CLK decrements the enabled counter by one until the full count cycle has been completed.

The initial count value of any counter can be changed by loading a new value into the counter's:

- LSB only (RL1, RL0 = 01),
- MSB only (RL1, RL0 = 10), or
- LSB first, and then MSB (RL1, RL0 = 11).

Read/Load Conditions RL1	RL0	Effect of Subsequent Write Operation
0	1	$\overline{WR}$ loads D7-D0 into LSB of counter selected by A1, A0.*
1	0	$\overline{WR}$ loads D7-D0 into MSB of counter selected by A1, A0.
1	1	First $\overline{WR}$ loads D7-D0 into LSB of counter selected by A1, A0. Next $\overline{WR}$ loads D7-D0 into counter's MSB.

*A1	A0	
0	0	Selects Counter 0
0	1	Selects Counter 1
1	0	Selects Counter 2

Figure 3. Initial Count Loading Summary

READING COUNT VALUES

The current status of a count sequence can be examined at any time by the program. This can be done either by reading the counter contents directly or by latching the counter contents into an auxiliary register and then reading that register.

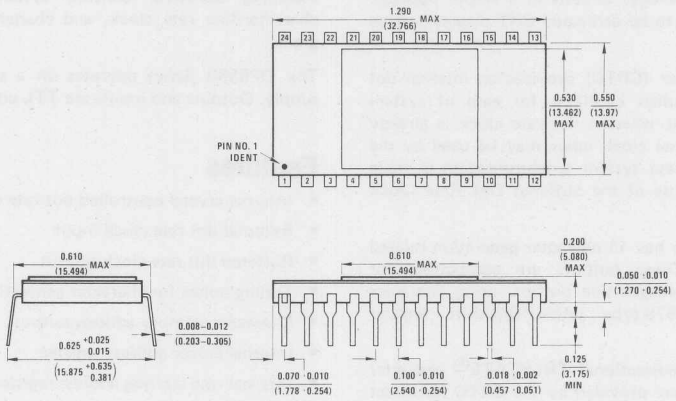
A counter can be read directly with the following bus conditions:

	$\overline{RD}$	A1	A0
To Read Counter 0	0	0	0
To Read Counter 1	0	0	1
To Read Counter 2	0	1	0

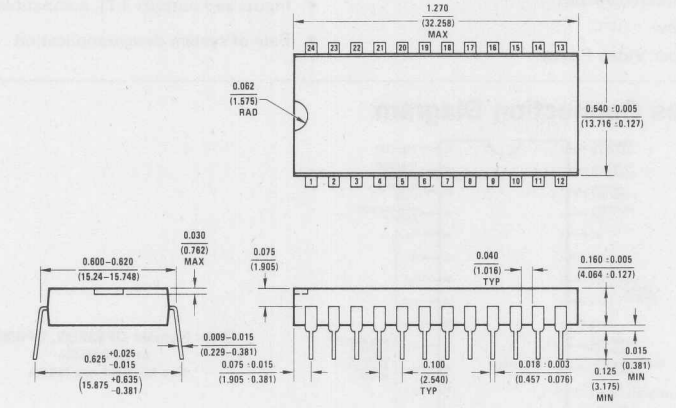
The count should remain stable during direct reading of a counter. Stability is assured by holding the Gate input low or inhibiting the CLK input (by means of external logic) for the duration of the read operation. Counter status can also be sampled without inhibiting the count sequence. This is done by issuing a control word to the counter with RL1, RL0 = 00, followed by an I/O read of that counter's location. The RL1, RL0 bits cause the contents of the addressed counter to be latched into the auxiliary register. The subsequent read operations access the auxiliary register.

When reading either a counter or the auxiliary register, the read operation must follow the format programmed for that counter by RL0 and RL1. Note that issuing a latch command of RL1, RL0 = 00 does not alter the previously programmed RL0 and RL1.

Physical Dimensions
inches (millimeters)



24-Lead Hermetic DIP (D)
NS Package Number D24A



24-Lead Molded DIP (N)
NS Package Number N24A

National Semiconductor

Microprocessor Support Circuits

DP8350 Series Programmable CRT Controllers

General Description

The DP8350 Series of CRT Controllers are single-chip bipolar (I^2L technology) circuits in a 40-pin package. They are designed to be dedicated CRT display refresh circuits.

The CRT Controller (CRTC) provides an internal dot rate crystal controlled oscillator for ease of system design. For systems where a dot rate clock is already provided, an external clock input may be used by the CRTC. In either case system synchronization is made possible with the use of the buffered Dot Rate Clock Output.

The DP8350 Series has 11 character generation related timing outputs. These outputs are compatible for systems with or without line buffers, using character ROMS, or DM8678-type latch/ROM/shift register circuits.

12 bits (4k) of bidirectional TRI-STATE® character memory addresses are provided by the CRTC for direct interface to character memory.

Three on-chip registers provide for external loading of the row starting address, cursor address, and top-of-page address.

A complete set of video outputs is available including cursor enable, programmable vertical blanking, programmable horizontal sync, and programmable vertical sync.

The DP8350 Series CRTC provides for a wide range of programmability using internal mask programmable ROMs:

- Character Field (both number of dots/character and number of scan lines/character)
- Characters per Row
- Character Rows per Video Frame

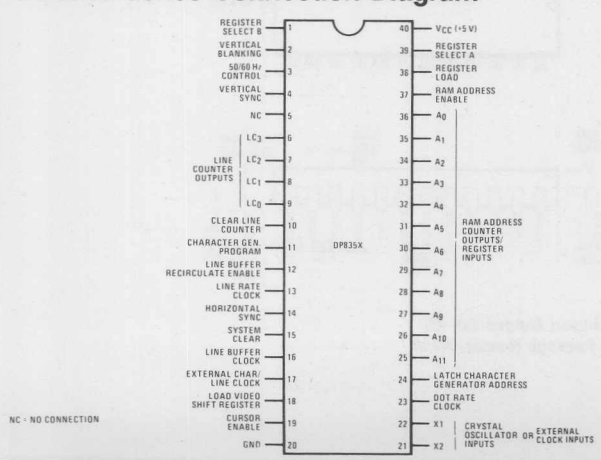
The CRTC also provides system sync and program inputs including 50/60 Hz control, system clear, external character/line rate clock, and character generator program.

The DP8350 Series operates on a single +5 V power supply. Outputs and inputs are TTL compatible.

Features

- Internal crystal controlled dot rate oscillator
- External dot rate clock input
- Buffered dot rate clock output
- Timing pulses for character generation
- Character memory address outputs (12 bits)
- Internal cursor address register
- Internal row starting address register
- Top-of-page address register (for scrolling)
- Programmable horizontal and vertical sync outputs
- Programmable cursor enable output
- Programmable vertical blanking output
- 50/60 Hz refresh rate
- Programmable characters/row (5 to 110)
- Programmable character field size (up to 16 dots x 16 scan line field size)
- Programmable character rows/frame (1 to 64)
- Single +5 V power supply
- Inputs and outputs TTL compatible
- Ease of system design/application

DP8350 Series Connection Diagram



Order Number DP8350N, DP8352N or DP8353N
See NS Package N40A

National Semiconductor

Preliminary

JULY 1978

INS8259 Programmable Interrupt Controller

General Description

The INS8259 is a Programmable Interrupt Controller chip contained in a standard 28-pin dual-in-line package. The chip, which is fabricated using N-channel silicon gate technology, functions as a versatile interrupt management device in a microcomputer system.

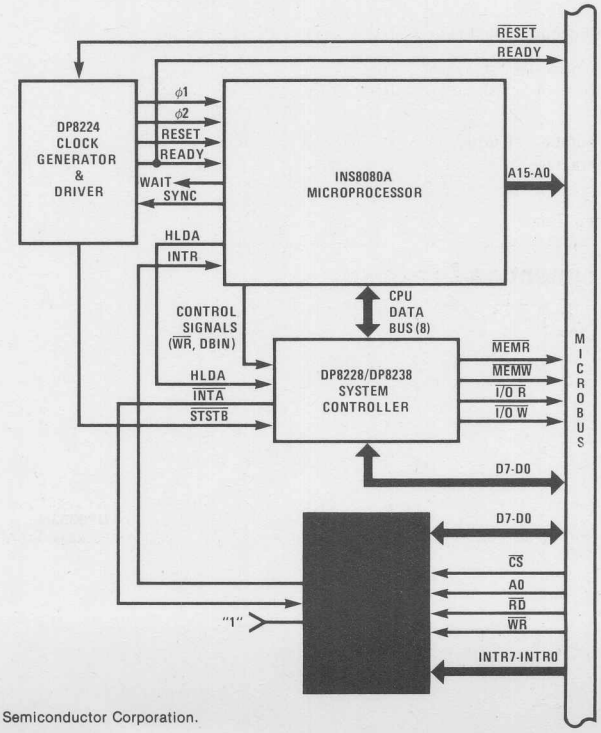
Each INS8259 resolves interrupt requests from up to eight different sources. These devices can be cascaded to provide interrupt management of up to 64 levels with no other circuitry required.

The INS8259 minimizes software and real time overhead when handling multi-level priority interrupts. Its four operating modes allow the device to satisfy diverse system requirements.

Features

- Performs Priority Control of up to Eight Interrupt Levels
- Can be Expanded to Handle up to 64 Levels Through Cascading
- Four Programmable Operating Modes
- Programmable Interrupt Vectors, Allowing Service Routines to be Located Anywhere in Memory
- Interrupt Request Inputs Can be Individually Masked
- TRI-STATE® TTL Drive Capability for Bidirectional Data and Control Buses
- Single +5 Volt Power Supply
- 28-Pin Dual-in-Line Package
- MICROBUS™* Compatible

INS8080 Family CPU Group to MICROBUS Configuration



*A trademark of National Semiconductor Corporation.

Appendix B

Additional Information Sources

Categories of sources are:

- Publications
- National's Microprocessor Users Group
- Training Centers
- Technical Support Program

Publications

Publications are available covering the various devices manufactured by National Semiconductor. The currently available literature is grouped in the following categories:

- Literature Index
- Handbooks
- Manuals
- Linear Applications, Vol I and II
- Databooks
- Guides
- Product Selection Guides
- Briefs
- Individual Application Notes
- Individual Data Sheets

Current Handbooks are:

- Audio Handbook
- Linear Applications Handbook Volume I and II
- SC/MP Microprocessor Applications Handbook
- Voltage Regulator Handbook
- Memory Applications Handbook
- Series-8000 Microprocessor Family Handbook

Current Databooks are:

- Linear Databook
- TTL Databook
- Interface Databook
- Memory Databook
- Special Function Databook
- MOS/LSI Databook
- CMOS Databook
- FET Databook
- IDM2900 Family Microprocessor Databook
- Discrete Databook
- Power Transistor Databook

Some Current Application Notes

Number

AN-114 Microprocessors - An Introduction
AN-128 Microprocessor Mates with MOS/LSI Keyboard Encoder
AN-142 Using a Microprocessor Beyond Apparent Speed
AN-159 Data Acquisition System Interface to Computers
AN-163 SC/MP Mates with Cassette Recorder
AN-164 A Data Concentrator Using PACE
AN-189 A PROM Programmer for the SC/MP LCDS
AN-197 Multi-processing with SC/MP
AN-198 Simplify CRT Terminal Design with the DP8350
AN-199 A Low Component Count Video Data Terminal Using the DP8350 CRT Controller and the INS8080 CPU
AN-201 Motor RPM Control Made Easy

Current Product Selection Guides

- Analog Switches Cross Reference Guide
- Analog Switches Selection Guide
- Audio Amplifier Comparison Guide
- Bipolar RAM Cross Reference Guide
- Bipolar PROM Cross Reference Guide
- Bipolar PROM/ROM Selection Guide
- Calculator Kits Guide
- CMOS Status/Cross Reference Guide
- CMOS Watch Circuits Guide
- DIP Resistor Array Cross Reference Guide
- Display Driver Selection Guide
- FET Op Amp Cross Reference Guide
- Fixed Voltage Regulator Guide
- Hybrid Packaging Capability Guide
- Industrial and Commercial Hybrid Op Amp Selection Guide
- Industrial Op Amp Selection Guide
- Interface Status Guide
- JFET Selection Guide
- JFET Cross Reference Guide
- LED Display Selection Guide
- LED Lamp Cross Reference Guide
- LED Lamp Selection Guide
- Linear Cross Reference Guide
- Linear Product Status Guide
- Military Op Amp Selection Guide
- Military Hybrid Op Amp Selection Guide
- MOS Product Status Guide
- RAM Cross Reference Guide
- RAM Selection Guide
- ROM Status Guide
- Special Purpose Drivers Guide
- Special Purpose Linear Circuits Guide
- TO-92 Transistor Guide
- Transducer Selection Guide
- TTL Status Guide
- TTL Functional Product Guide
- Variable Voltage Regulator Guide
- Voltage Comparator Guide
- 54C/74C Users Guide

Current briefs are:

- Linear Briefs
- Digital Briefs
- MOS Briefs
- Microprocessor Briefs
- FET Briefs
- Opto Briefs
- Transducer Briefs

For list of other available literature, refer to your local National Semiconductor Sales Representative.

Ordering Information

All orders must be prepaid. Domestic orders must be accompanied by a check or a money order made payable to National Semiconductor Corp.; orders destined for shipment outside of the U.S. must be accompanied by U.S. funds. Orders will be shipped by postage-paid Third Class mail. Please allow approximately six to eight weeks for domestic delivery, longer for delivery outside the U.S.

The Data Bookshelf:

Tools for the Design Engineer

National Semiconductor's Data Bookshelf is a compendium of information about a product line unmatched in its breadth in the industry. The independent volumes that comprise the Bookshelf — over 5,000 pages — describe in excess of 6,000 solid-state devices; devices that span the entire spectrum of semiconductor processes, and that range from the simplest of discrete transistors to microprocessors — those most sophisticated marvels of modern integrated circuit technology.

Active and passive devices and circuits; hybrid and monolithic structures; discrete and integrated components ... complete electrical and mechanical specifications; charts, graphs, and tables, test circuits and waveforms; design and application information ... Whatever you need you'll find it in the designer's ultimate reference source — National Semiconductor's Data Bookshelf.